



**TUGAS AKHIR - TE 145561**

**RANCANG BANGUNG ALAT PENGHITUNG PENDAPATAN  
KOIN PADA LOKER SUSUN BERBASIS MIKROKONTROLER**

Ega Hasbi Rizqullah  
NRP 10311500000016

Dosen Pembimbing  
Ir. Hany Boedinugroho, MT.

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**FINAL PROJECT - TE 145561**

***CONSTRUCTION COUNTER COIN DEVICE OF STACKING  
LOCKERS BASED ON MICROCONTROLLER***

Ega Hasbi Rizullah  
NRP 10311500000016

Supervisor  
Ir. Hany Boedinugroho, MT.

COMPUTER CONTROL STUDY PROGRAM  
Electrical and Automation Engineering Department  
Vocational Faculty  
Sepuluh Nopember Institute of Technology  
Surabaya 2018




## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **"Rancang Bangun Alat Penghitung Pendapatan Koin pada Loker Susun Berbasis Mikrokontroler"** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 24 Juli 2018



Ega Hasbi Rizqullah  
NRP 10311500000016

-----Halaman ini sengaja dikosongkan-----

**RANCANG BANGUNG ALAT PENGHITUNG  
PENDAPATAN KOIN PADA LOKER SUSUN BERBASIS  
MIKROKONTROLER**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Ahli Madya Teknik  
Pada  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember**



**Menyetujui :**

**Dosen Pembimbing**

**Ir. Hany Boedinugroho, MT.**  
**NIP. 19610706 198701 1 001**

**SURABAYA  
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----



# **RANCANG BANGUNG ALAT PENGHITUNG PENDAPATAN KOIN PADA LOKER SUSUN BERBASIS MIKROKONTROLER**

**Nama : Ega Hasbi Rizqullah**  
**Pembimbing : Ir. Hany Boedinugroho, MT.**

## **ABSTRAK**

Pada era globalisasi ini teknologi sudah sangat canggih dimana sudah berbentuk digital dan sistem otomatis. Perangkat yang digunakan sudah berskala kecil ataupun sering disebut berskala mikro. Hal tersebut juga dapat memberikan inovasi pada penyewaan loker dimana perhitungan pendapatan yang dilakukan secara manual terbilang masih belum terhitung secara presisi dikarenakan pengamatan lama waktu penyewaan loker yang dilakukan pihak penyewa masih mengamati langsung loker yang digunakan oleh pengguna.

Dalam Proyek Akhir ini perancang akan membuat sistem berupa mikrokontroler yang akan menghitung setiap koin yang masuk menggunakan *Coin Acceptor* dan lama waktu penggunaan setiap loker yang akan mempengaruhi jumlah pembayaran. Cara kerja alat ini adalah pengguna akan membayar harga yang tertera pada LCD tergantung dari lama waktu penggunaan loker sebelum mengambil barang. Kemudian setelah pengguna membayarkan koin tersebut maka sistem tersebut akan menginformasikan koin yang terbayar setiap waktu penggunaan.

Hasil dari Tugas Akhir ini adalah alat dapat membaca koin yang masuk berupa uang koin Rp 500 kuning dan perak serta dapat melakukan perhitungan tarif harga berdasarkan lama waktu dari penggunaan sewa loker. Pada alat ini memiliki presentase setelah dilakukan 20 kali percobaan pada setiap loker menghasilkan keberhasilan 91,25% dengan presentase error sebesar 8,75%.

**Kata Kunci :** Loker, *Coin Acceptor*, *Coin*, *Timer*, *LCD*, Pembayaran, Mikrokontroler.

-----Halaman ini sengaja dikosongkan-----

## **CONSTRUCTION COUNTER COIN DEVICE OF STACKING LOCKERS BASED ON MICROCONTROLLER**

**Name : Ega Hasbi Rizqullah**  
**Supervisor : Ir. Hany Boedinugroho, MT.**

### **ABSTRACT**

*In this era of globalization technology has been very sophisticated which has been shaped digital and automated systems. The devices used are small scale or often called micro scale. It can also provide an innovation on locker rental where the calculation of revenues done manually is still not precisely calculated due to the observation of the duration of rental tenant locker is still observing the locker directly used by the user.*

*In this Final Project the designer will create a microcontroller system that will calculate every incoming coin using Coin Acceptor and the duration of use of each locker that will affect the amount of payment. How this tool works is that users will pay the price listed on the LCD depending on the length of time the use of lockers before taking ba-rang. Then after the user pays the coin then the system will inform the coin payable every time usage.*

*The results of this Final Project is a tool to read coins that come in the form of coins Rp 500 yellow and silver and can perform the calculation of price rates based on the length of time from the use of rent lockers. In this tool has a percentage after 20 attempts on each locker to produce 91.25% success with a percentage error of 8.75%.*

**Keywords : Locker, Coin Acceptor, Coin, Timer, LCD, Payment, Microcontroller.**

-----Halaman ini sengaja dikosongkan-----

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

### **RANCANG BANGUNG ALAT PENGHITUNG PENDAPATAN KOIN PADA LOKER SUSUN BERBASIS MIKROKONTROLER**

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Ir. Hany Boedinugroho, MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 24 Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

	HALAMAN
HALAMAN JUDUL .....	i
PERNYATAAN KEASLIAN TUGAS AKHIR .....	v
HALAMAN PENGESAHAN .....	vii
ABSTRAK .....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
<b>1. BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Metodologi Penelitian .....	3
1.6 Sistematika Laporan .....	4
1.7 Relevansi .....	5
<b>2. BAB II TEORI DASAR .....</b>	<b>7</b>
2.1 Pendapat .....	7
2.2 Mikrokontroler Atmega 16 .....	8
2.3 ATmega 8 .....	10
2.4 <i>Coin Acceptor</i> .....	14
2.5 <i>Power Supply</i> .....	15
2.6 RTC ( <i>Real Time Clock</i> ) .....	18
2.7 <i>Liquid Cristal Display</i> (LCD) 20x4 .....	19
2.8 XL4005 .....	20
2.9 USB to TTL Serial .....	21
2.10 CodeVisonAVR .....	22
2.11 Bahasa C .....	24
2.12 PLX-DAQ .....	25
2.13 Khazama .....	29
<b>BAB III PERENCANAAN DAN PEMBUATAN ALAT .....</b>	<b>33</b>

3.1	Blok Fungsional Sistem .....	33
3.2	Cara Kerja Alat .....	34
3.3	Perancangan <i>Hardware</i> .....	35
3.3.1	<i>Setting Coin Acceptor CH-923</i> .....	35
3.3.2	Perancangan RTC( <i>Real Time Clock</i> ) DS1307 .....	37
3.3.3	Perancangan USB to TTL Serial .....	38
3.3.4	Konfigurasi LCD dengan Mikrokontroler Atmega 16..	38
3.4	Perancangan <i>Software</i> .....	39
3.4.1	Flowchart Pembayaran .....	42
3.5	Desain Kubus Kontroler .....	43
3.6	Desain Loker .....	46
<b>4</b>	<b>BAB IV PENGUKURAN DAN PENGUJIAN .....</b>	<b>49</b>
4.1	Pengukuran Rangkaian <i>Power Supply</i> .....	49
4.2	Pengujian LCD ( <i>Liquid Crystal Display</i> ) 20x4 .....	50
4.3	Pengujian RTC DS1307 .....	51
4.4	Pengujian Komunikasi USB to TTL Serial .....	52
4.5	Pengujian Sensor <i>Coin Acceptor CH-923</i> .....	53
4.6	Petunjuk Penggunaan Alat .....	55
4.7	Pengujian Keseluruhan Sistem Alat .....	63
4.7.1	Fitur <i>Timer</i> dan Tarif Penggunaan .....	63
4.7.2	Fitur <i>Insert Coin</i> .....	64
4.7.3	Fitur <i>Serial Data</i> .....	66
<b>5</b>	<b>BAB V PENUTUP .....</b>	<b>71</b>
5.1	Kesimpulan .....	71
5.2	Saran .....	71
	<b>DAFTAR PUSTAKA .....</b>	<b>73</b>
	<b>LAMPIRAN .....</b>	<b>75</b>
	<b>RIWAYAT HIDUP PENULIS .....</b>	<b>101</b>



## DAFTAR GAMBAR

	HALAMAN
Gambar 2. 1 Konfigurasi Mikrokontroler ATmega16 .	10
Gambar 2. 2 Susunan Pin ATmega8	11
Gambar 2. 3 <i>Coin Acceptor</i> .....	15
Gambar 2. 4 Diagram Blok <i>Power Supply</i> .....	15
Gambar 2. 5 <i>Trafo Step Down</i> .....	16
Gambar 2. 6 <i>Rectifier</i> pada <i>Power Supply</i> .	17
Gambar 2. 7 <i>Filter</i> pada <i>Power Supply</i> .....	17
Gambar 2. 8 RTC DS1307 .....	18
Gambar 2. 9 LCD 20x4 .....	20
Gambar 2. 10 XL4005 .....	21
Gambar 2. 11 Modul USB to TTL Serial .....	22
Gambar 2. 12 Menampilkan “Developer Tab” .....	26
Gambar 2. 13 Mengaktifkan Makro .....	26
Gambar 2. 14 Contoh Program Arduino .....	27
Gambar 2. 15 Konfigurasi PLX – DAQ.....	28
Gambar 2. 16 PIX-DAQ Berhasil Mengirim Data .....	28
Gambar 2. 17 Read chip signature .....	29
Gambar 2. 18 Pemberitahuan <i>Chip Signature</i> .....	29
Gambar 2. 19 Memasukkan File.hex .....	30
Gambar 2. 20 Pilih <i>File</i> .....	30
Gambar 2. 21 <i>File.hex</i> berhasil di <i>download</i> .....	30
Gambar 2. 22 Memilih <i>Manual</i> .....	31
Gambar 2. 23 Program Berhasil di <i>Download</i> .....	31
 Gambar 3. 1 Blok Fungsional Sistem.....	 33
Gambar 3. 2 Rangkaian RTC DS1307 .....	37
Gambar 3. 3 Rangkaian USB to TTL.....	38
Gambar 3. 4 Rangkaian LCD 20x4 dengan mikrokontroler .....	38
Gambar 3. 5 <i>Flowchart</i> proses awal menyewa loker .....	40
Gambar 3. 6 <i>Flowchart</i> proses pengambilan barang pada loker .....	41
Gambar 3. 7 <i>Flowchart</i> Perhitungan Tarif .....	42
Gambar 3. 8 <i>Flowchart</i> Pembayaran <i>Coin</i> .....	43
Gambar 3. 9 Desain Kubus Kontroler .....	44
Gambar 3. 10 Kubus Kontroler Tampak Depan.....	44

Gambar 3. 11 Kubus Kontroler Tampak Samping .....	45
Gambar 3. 12 Kubus Kontroler Tampak Atas .....	45
Gambar 3. 13 Desain Loker .....	46
Gambar 3. 14 Hasil Desain Loker Depan .....	47
Gambar 3. 15 Hasil Desain Loker Samping .....	47
Gambar 3. 16 Hasil Desain Loker Samping .....	48
Gambar 3. 17 Hasil Rancangan Desain Keseluruhan .....	48
 Gambar 4. 1 Pengujian tampilan dari LCD .....	 51
Gambar 4. 2 Komunikasi Serial Menggunakan USB to TTL pada Excel .....	52
Gambar 4. 3 Petunjuk Penggunaan .....	55
Gambar 4. 4 Kondisi Loker .....	56
Gambar 4. 5 Buka Loker .....	56
Gambar 4. 6 Taruh Barang .....	57
Gambar 4. 7 <i>Set Password</i> .....	57
Gambar 4. 8 Masukkan <i>Password</i> .....	58
Gambar 4. 9 Hapus <i>Password</i> .....	58
Gambar 4. 10 Simpan <i>Password</i> .....	59
Gambar 4. 11 Petunjuk Penggunaan .....	59
Gambar 4. 12 Pilih Loker .....	60
Gambar 4. 13 <i>Enter Password</i> .....	60
Gambar 4. 14 Hapus <i>Password</i> .....	61
Gambar 4. 15 Tekan # .....	61
Gambar 4. 16 Notif Tarif Pembayaran .....	62
Gambar 4. 17 Masukkan <i>Coin</i> .....	62
Gambar 4. 18 Jenis <i>Coin</i> .....	63
Gambar 4. 19 Total Waktu dan Tarif Penggunaan .....	64
Gambar 4. 20 Jumlah Pembayaran <i>Coin</i> .....	65
Gambar 4. 21 Pembayaran <i>Coin</i> .....	65
Gambar 4. 22 Data Terkirim Pada Excel .....	66

## DAFTAR TABEL

HALAMAN

Tabel 4. 1 Tabel pengukuran tegangan <i>Power Supply</i>	49
Tabel 4. 2 Data Pengujian <i>Power Supply</i>	50
Tabel 4. 3 Hasil Modul RTC ( <i>Real Time Clock</i> )	51
Tabel 4. 4 Hasil Pengujian <i>Coin Acceptor</i> Dengan <i>Coin</i> Rp 500 kuning	53
Tabel 4. 5 Hasil Pengujian <i>Coin Acceptor</i> Dengan <i>Coin</i> Rp 500 perak	53
Tabel 4. 6 Hasil Pengujian <i>Coin Acceptor</i> Dengan <i>Coin</i> Rp 200 perak	54
Tabel 4. 7 Hasil Pengujian <i>Coin Acceptor</i> Dengan <i>Coin</i> Rp 1000 perak	54
Tabel 4. 8 Tabel Pengujian Kinerja Alat Keseluruhan pada Loker A ....	67
Tabel 4. 9 Tabel Pengujian Kinerja Alat Keseluruhan pada Loker B ....	67
Tabel 4. 10 Tabel Pengujian Kinerja Alat Keseluruhan pada Loker C ...	68
Tabel 4. 11 Tabel Pengujian Kinerja Alat Keseluruhan pada Loker D...	69

-----Halaman ini sengaja dikosongkan-----

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi dari tahun ke tahun selalu mengalami perkembangan yang sangat pesat. Banyaknya fasilitas kemudahan - kemudahan yang ditimbulkan oleh perkembangan teknologi secara langsung berdampak kepada kehidupan manusia. Berkembangnya ilmu pengetahuan dan teknologi merupakan ciri dari perkembangan zaman. Salah satu contoh perkembangan teknologi tersebut adalah dalam bidang mikrokontroler. Perkembangan sistem berbasis mikrokontroler pada saat ini berkembang dengan sangat pesat, mulai dari sistem yang sederhana hingga sistem yang sangat fatal bagi sebuah perusahaan. Pada era kemajuan teknologi, kini banyak mahasiswa berlomba - lomba dalam menciptakan inovasi teknologi demi menunjang dalam membantu kebutuhan manusia. Perkembangan teknologi pada saat ini mencakup banyak sekali aspek kehidupan. Salah satunya adalah aspek keamanan. Maraknya pencurian barang hingga pembobolan tempat penyimpanan barang mendorong masyarakat berlomba - lomba untuk berinovasi membuat suatu sistem pengamanan untuk mengurangi peluang bagi para pencuri tersebut mengambil barang milik orang lain. Salah satu contoh dari perkembangan sistem berbasis mikrokontroler saat ini yaitu sistem keamanan pada loker yang biasa terdapat pada tempat penyewaan loker yang tersedia di tempat - tempat umum. Loker merupakan tempat penyimpanan barang dimana biasa dipakai pada tempat-tempat wisata, perpustakaan, tempat olahraga ataupun tempat umum lainnya. Fungsi loker sebagai tempat penyimpanan seharusnya memiliki tingkat keamanan tinggi karena yang disimpan di dalamnya adalah barang - barang yang berharga. Keamanan sebuah loker sangat bergantung pada kunci pintunya. Pada dasarnya penitipan barang pada loker masih dilakukan secara konvensional yaitu ketika pengguna akan menitipkan barang pada loker maka pengguna harus mengisikan identitas terlebih dahulu ataupun pengguna akan diberi kunci loker untuk membuka dan menutup lokernya sendiri. Dalam hal ini sering kali terjadi kelalaian pada pihak penyewa loker dalam melakukan penjagaan loker. Salah satu kenyataannya adalah sering terjadinya pencurian dan kehilangan barang pada tempat penyewaan loker. Para pencuri dengan mudahnya dapat membuka pengunci

loker menggunakan seutas kawat atau dengan kunci tiruan lainnya. Selain itu kunci konvensional dapat dengan mudah untuk digandakan, rusak bahkan ada kemungkinan hilang atau lupa mengunci pintu loker. Selain itu sistem pembayaran pada penyewaan loker tersebut dinilai masih kurang efektif dikarenakan harus melibatkan orang dalam melakukan transaksi pembayaran sewa loker tersebut.

Dikarenakan sistem pembayaran loker masih menggunakan secara manual yaitu dengan cara penyedia persewaan loker masih melakukan pengamatan secara langsung pada lama waktu penggunaan loker dan dinilai masih belum presisi dalam perhitungan harga setiap lama penggunaan loker. Maka dari itu dibuatlah sebuah inovasi penghitungan koin yang masuk ke dalam loker yang akan terhitung secara otomatis dan harga yang terhitung dipengaruhi oleh lama waktu penggunaan setiap loker tanpa harus pihak penyewa loker mengamati setiap loker tersebut dan juga dapat memberikan informasi nilai pendapatan dari setiap loker.

## **1.2 Permasalahan**

Adapun permasalahan yang mendasari pembuatan Tugas Akhir ini adalah loker yang digunakan di tempat – tempat umum pada saat ini masih menggunakan sistem pembayaran yang manual dimana harus terdapat petugas untuk menjaga tempat pembayaran untuk yang dilakukan pengguna sebagai tempat melakukan pembayaran. Seorang petugas juga masih melakukan perhitungan waktu secara manual dari lama penggunaan loker yang dapat menimbulkan masalah apabila terjadi salah dalam perhitungan lama waktu sewa loker dikarenakan kurang keakuratan dalam perhitungan lama waktu sewa yang dapat berpengaruh pada harga tarif dari setiap sewa loker.

## **1.3 Batasan Masalah**

Pada tugas akhir ini, memiliki batasan-batasan masalah yang diambil, diantaranya :

1. Alat pembayaran hanya berupa uang koin kuning dan perak Rp 500.
2. Alat tidak dapat melakukan kembalian uang apabila terjadi kelebihan.

#### 1.4 Tujuan

Tujuan yang akan dicapai dari Program Tugas Akhir ini adalah mendeteksi koin yang masuk ke dalam *Coin Acceptor* sebagai alat pembayaran loker dan menghitung koin yang terkumpul dari lama waktu penggunaan pada setiap loker.

#### 1.5 Metodologi Penelitian

Dalam pelaksanaan tugas akhir berupa Rancang Bangun Alat Penghitung Pendapatan pada Loker Susun Berbasis Mikrokontroler Ada beberapa tahap yang perlu dipersiapkan yaitu sebagai berikut:

- **Pengamatan Permasalahan**

Pengamatan dilakukan untuk mengkalibrasi setiap koin yang digunakan akan berpengaruh pada program yang akan dibuat pada mikrokontroler sehingga dapat dilakukan perhitungan nilai koin yang masuk kedalam mikrokontroler untuk menghindari nilai kesalahan dalam pengaplikasian program.

- **Studi Literatur**

Kegiatan ini dilakukan dengan cara mencari teori penunjang dari buku, jurnal, dan artikel dari media online maupun media cetak. Literatur berguna untuk menunjang dalam penyelesaian masalah yang sedang dihadapi. Dalam menghadapi masalah diperlukan sumber ilmiah agar data perhitungan yang diberikan sudah teruji dan dapat dipertanggung jawabkan. Teori yang dicari berkaitan dengan bahasa pemrograman serta perhitungan mengenai cara penggunaan modul *coin acceptor*.

- **Perancangan Alat**

Pada tahapan ini dilakukan perancangan alat setelah melewati tahapan dari sebelumnya. Dimulai dengan membuat rancang bangun pada sistem alat, didesain sedemikian rupa sesuai dengan yang telah direncanakan. Peletakan *Coin Acceptor* dibuat secara presisi agar hasil sesuai dengan yang diinginkan. Kemudian dibuat pemrograman pada mikrokontroler yang dihubungkan pada rancang bangun yang telah dibuat. Pemrograman diatur

sesuai dengan algoritma dari perhitungan nilai koin yang masuk. Pengguna yang akan mengambil barang pada loket tersebut diharuskan membayar pada tempat pembayaran menggunakan koin terlebih dahulu baru dapat mengambil barang.

- **Pengujian Alat dan Analisis Data**

Setelah pembuatan rancangan selesai, selanjutnya akan dilakukan pengujian alat secara langsung. Hal tersebut bertujuan untuk mengetahui apakah rancangan yang dibuat telah sesuai dengan apa yang direncanakan. Pada tahap ini juga akan dilakukan pengambilan data. Pengambilan data bertujuan untuk mengetahui apakah data yang dihasilkan sudah sesuai dengan data yang diinginkan sehingga apabila terjadi kesalahan pada alat maka kesalahan tersebut dapat diselesaikan dengan baik.

- **Kesimpulan**

Pada tahap ini akan menyimpulkan dari alat yang telah dibuat sudah sesuai dengan yang diharapkan. Bagian kesimpulan berpacu pada data pengujian alat sehingga dapat dilakukan analisa dari hasil data yang telah didapat untuk diberikan tindakan pada tahap selanjutnya.

- **Penyusunan Laporan Akhir**

Tahap terakhir yang perlu dilakukan adalah penyusunan laporan akhir yang bertujuan sebagai bukti tertulis bahwa pernah dilakukan penelitian mengenai alat tersebut dengan menggunakan metode yang ditetapkan. Laporan juga bertujuan untuk memberikan referensi pada penelitian berikutnya.

## **1.6 Sistematika Laporan**

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:



<b>Bab I</b>	<b>Pendahuluan</b> Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan dan relevansi.
<b>Bab II</b>	<b>Teori Dasar</b> Bab ini menjelaskan tentang tinjauan pustaka dari <i>Power Supply</i> , <i>Coin Acceptor</i> , LCD 20x4, <i>Real Time Clock</i> , USB to TTL Serial, <i>software</i> yang digunakan adalah CodeVision AVR,
<b>Bab III</b>	<b>Perancangan dan Pembuatan Alat</b> Bab ini membahas perencanaan dan pembuatan perangkat keras ( <i>Hardware</i> ) yang meliputi desain alat dan aktuator yang digunakan dan pembuatan perangkat lunak ( <i>Software</i> ) yang meliputi program pada CodeVision AVR untuk menjalankan alat tersebut.
<b>Bab IV</b>	<b>Pengukuran dan Pengujian</b> Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian <i>Coin Acceptor</i> , DS1307 dan LCD, pengujian <i>Coin</i> untuk mengetahui keaktifan alat yang digunakan, pengujian rangkaian LCD untuk mengaktifkan dari DS1307.
<b>Bab V</b>	<b>Penutup</b> Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

## 1.7 Relevansi

- Mengoptimalkan setiap koin yang masuk terhitung oleh mikrokontroler yang digunakan.
- Menghasilkan sebuah sistem yang dapat menghitung koin yang telah diterima dari setiap loker oleh pengguna.

-----Halaman ini sengaja dikosongkan-----

## **BAB II**

### **TEORI DASAR**

Bab ini membahas mengenai teori dasar dari peralatan yang digunakan dalam Tugas Akhir yang berjudul Rancang Bangun Alat Penghitung Pendapatan Koin pada Loker Susun Berbasis Mikrokontroler yang meliputi 11 poin. Materi ini digunakan sebagai dasar materi untuk pembuatan alat yang dibuat masing-masing mahasiswa untuk pembuatan keseluruhan alat ini.

#### **2.1 Pendapatan**

Pendapatan adalah seluruh penerimaan baik berupa uang maupun berupa barang yang berasal dari pihak lain maupun hasil industri yang dinilai atas dasar sejumlah uang dari harta yang berlaku saat itu. Pendapatan merupakan sumber penghasilan seseorang untuk memenuhi kebutuhan sehari – hari dan sangat penting artinya bagi kelangsungan hidup dan penghidupan seseorang secara langsung mau pun tidak langsung (Suroto, 2000).

Untuk memahami arti dari pendapatan, maka akan diuraikan pengertian dari pendapatan itu sendiri. Menurut Ikatan Akuntansi Indonesia (2009) dalam buku Standart Akuntansi Keuangan menyebutkan bahwa pendapatan adalah: “Arus masuk bruto dari manfaat ekonomi yang timbul dari aktivitas normal perusahaan selama satu periode, bila arus masuk itu mengakibatkan kenaikan ekuitas, yang tidak berasal dari kontribusi penanaman modal”.

Sedangkan menurut Accounting Principle Board dikutip oleh Theodorus Tuanakotta (1984:153) dalam buku Teori Akuntansi pengertian pendapatan adalah” Pendapatan sebagai inflow of asset kedalam perusahaan sebagai akibat penjualan barang dan jasa”.

Menurut pendapat lain, pendapatan adalah kenaikan kotor dalam asset atau penurunan dalam lialibilitas atau gabungan dari keduanya selama periode yang dipilih oleh pernyataan pendapatan yang berakibat dari investasi yang halal, keuntungan, seperti manajemen rekening investasi terbatas (Suroto, 2000).

Pendapatan merupakan suatu unsure yang harus dilakukan dalam melakukan suatu usaha karena dalam melakukan suatu usaha tentu ingin mengetahui nilai atau jumlah pendapatan yang diperoleh selama melakukan usaha. Menurut Sumitro Joyohadikusumo (1957) Pendapatan merupakan jumlah barang dan jasa yang memenuhi

tingkat hidup masyarakat, dimana dengan adanya pendapatan yang dimiliki oleh setiap jiwa disebut dengan pendapatan perkapita dimana pendapatan perkapita menjadi tolak ukur kemajuan atau perkembangan ekonomi (Heri, 2015).

Pendapatan sangat berpengaruh bagi kelangsungan suatu usaha, semakin besar pendapatan yang diperoleh maka semakin besar kemampuan suatu usaha untuk membiayai segala pengeluaran dan kegiatan – kegiatan yang akan dilakukan. Kondisi seseorang dapat diukur dengan menggunakan konsep pendapatan yang menunjukkan jumlah seluruh uang yang diterima oleh seseorang atau rumah tangga selama jangka waktu tertentu (Samuelson dan Nordhaus, 2013) Ada definisi lain mengenai pendapatan yaitu pendapatan dikatakan sebagai jumlah penghasilan yang diperoleh dari hasil pekerjaan dan biasanya pendapatan seseorang dihitung setiap tahun atau setiap bulan (Irman, 2013).

## **2.2 Mikrokontroler Atmega 16**

Mikrokontroler merupakan suatu terobosan teknologi mikroprosesor dan mikrokomputer yang merupakan teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang sangat kecil. Mikrokontroler merupakan sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan PC (Personal Computer) yang memiliki beragam fungsi (Heri, 2015).

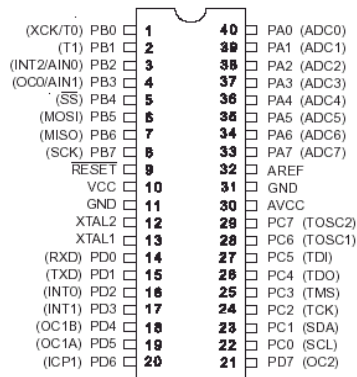
Tidak seperti sistem komputer yang mampu menangani berbagai macam program aplikasi. mikrokontroler hanya bisa digunakan untuk suatu aplikasi tertentu saja, perbedaan lainnya terletak pada perbandingan RAM (Random Access Memory) dan ROM (Read Only Memory) dimana program-program pengguna disimpan dalam ruang RAM (Random Access Memory) yang relatif besar, sedangkan prosedur antarmuka perangkat keras disimpan dalam ruang ROM (Read Only Memory) yang kecil. Sedangkan pada perbandingan mikrokontroler ROM (Read Only Memory) dan RAM (Random Access Memory) yang besar, artinya program kontrol disimpan dalam ROM (Read Only Memory) pada bias Masked ROM (Read Only Memory) atau Flash PEROM yang ukurannya relatif lebih besar, sedangkan RAM (Random Access Memory) digunakan sebagai tempat penyimpanan sementara,

termasuk register-register yang digunakan pada mikrokontroler yang bersangkutan.

Mikrokontroler ATmega16 merupakan suatu sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu chip IC, sehingga sering juga disebut dengan single chip mikrokomputer. Mikrokontroler biasa dikelompokkan dalam satu keluarga, masing-masing mikrokontroler mempunyai spesifikasi tersendiri namun masih kompatibel dalam pemrogramannya (Irman, 2013).

Didalam pembuatan Tugas Akhir ini penulis memilih mikrokontroler AVR ATmega 16 sebagai prosesor dari alat yang akan dibuat. AVR merupakan seri mikrokontroler CMOS 8 bit buatan Atmel, berbasis arsitektur RISC (Reduced Instruction Set Computer). Atmel merupakan salah satu vendor yang bergerak dibidang mikroelektronika, telah mengembangkan AVR (Alf and Vegard's Risc processor) sekitar tahun 1997. Berbeda dengan mikrokontroler MCS51, AVR menggunakan arsitektur RISC (Reduce Instruction Set Computer) yang mempunyai lebar bus data 8 bit, perbedaan ini bisa dilihat dari frekuensi kerjanya. MCS51 memiliki frekuensi kerja 1/12 kali frekuensi osilator sedangkan frekuensi kerja AVR sama dengan frekuensi osilator. Jadi dengan frekuensi osilator yang sama, kecepatan AVR 12 kali lebih cepat dibanding kecepatan MCS51 (Irman, 2013).

Secara umum AVR dibagi menjadi 4 kelas, yaitu Attiny, AT90Sxx, ATmega dan AT86RFxx. Perbedaan antar tipe AVR terletak pada fitur-fitur yang ditawarkan, sementara dari segi arsitektur dan set instruksi yang digunakan hampir sama. Masing – masing portnya memiliki input dan output sebanyak 32 buah dengan resolusi 8 bit ADC. Semua pinnya memiliki tegangan 3,3 V sampai 5 V. Untuk pin ADC 0 sampai 5 V. Masing – masing port pada ATmega mampu mengeluarkan arus hingga 20mA dan maksimum mengeluarkan arus 200mA. Konfigurasi pin mikrokontroler Atmega16 dengan kemasan 40 pin dapat dilihat pada Gambar 2.1. Dari gambar tersebut dapat terlihat ATmega16 memiliki 8 pin untuk masing-masing port A, port B, port C, dan port D.



**Gambar 2. 1** Konfigurasi Mikrokontroler ATmega16 (Irman, 2013).

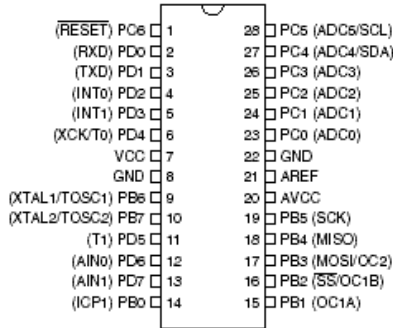
Adapun fitur Mikrokontroler ATmega16 kapabilitas detail dari adalah sebagai berikut :

1. Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi, dengan daya rendah.
2. Arsitektur RISC dengan throughput mencapai 16 MIPS pada frekuensi 16MHz.
3. Memiliki kapasitas Flash memori 16 KByte, EEPROM 512 Byte dan SRAM 1 KByte.
4. Saluran I/O sebanyak 32 buah, yaitu port A, port B, port C, dan port D.
5. CPU yang terdiri atas 32 buah register.
6. Unit interupsi internal dan eksternal.
7. Port USART untuk komunikasi serial.
8. Fitur Peripheral.

### 2.3 ATmega 8

ATMEGA 8 adalah mikrokontroler CMOS 8-bit daya rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus clock, ATMEGA 8 mempunyai throughput mendekati 1 MPS per MHz membuat disain dari sistem untuk mengoptimasi konsumsi daya versus kecepatan proses (John, 2007).

Susunan pin – pin dari IC mikrokontroler ATMEGA 8 diperlihatkan pada Gambar 2.2. IC ini tersusun dari 28 pin yang memiliki beberapa fungsi tertentu.



**Gambar 2. 2** Susunan Pin ATmega8 (Irman, 2013)

ATMega8 memiliki 28 pin yang masing – masing pin – nya memiliki fungsi yang berbeda – beda baik sebagai port ataupun sebagai fungsi yang lain. Berikut akan dijelaskan tentang kegunaan dari masing – masing kaki pada ATMega8.

1. VCC  
Merupakan supply tegangan untuk digital.
2. GND  
Merupakan ground untuk semua komponen yang membutuhkan grounding.
3. Port B  
Adalah 8 buah pin mulai dari pin B.0 sampai dengan pin B.7. Tiap pin dapat digunakan sebagai input dan juga output. Port B merupakan sebuah. 8-bit bit-directional I/O port dengan internal pull-up resistor. Sebagai input, pin – pin yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika pull-up resistor diaktifkan. Jika ingin menggunakan tambahan kristal, maka cukup untuk menghubungkan kaki dari kristal ke kaki pada pin port B. Namun jika tidak digunakan, maka cukup untuk dibiarkan saja. Pengguna kegunaan dari masing – masing kaki ditentukan dari clock fuse setting-nya.

4. Port C

Port C merupakan sebuah 7-bit bi-directional I/O yang di dalam masing – masing pin terdapat pull-up resistor. Jumlah pin-nya hanya 7 buah mulai dari C.0 sampai dengan pin C.6. Sebagai keluaran / output, port C memiliki karakteristik yang sama dalam hal kemampuan menyerap arus ( sink ) ataupun mengeluarkan arus ( source).

5. Reset / PC6

Jika RSTDISBL Fuse diprogram, maka PC6 akan berfungsi sebagai pin I/O. Untuk diperhatikan juga bahwa pin ini memiliki karakteristik yang berbeda dengan pin – pin yang terdapat pada port C. Namun jika RSTDISBL Fuse tidak deprogram, maka pin ini akan berfungsi sebagai input reset. Dan jika level tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun clock-nya tidak berkerja.

6. Port D

Port D merupakan 8-bit bi-directional I/O dengan internal pull-up resistor. Fungsi dari port ini sama dengan port – port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada port ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

7. AVCC

Pada pin ini memiliki fungsi sebagai power supply tegangan untuk ADC. Untuk pin ini harus dihubungkan secara terpisah dengan VCC karena pin ini digunakan untuk analog saja. Bahkan jika ACD pada AVR tidak digunakan, tetap saja disarankan untuk menghubungkan secara terpisah dengan VCC. Cara menghubungkan AVCC adalah melewati low-pass filter setelah itu dihubungkan dengan VCC.

8. AREF

Merupakan pin referensi analog jika menggunakan ADC. Pada AVR status Register mengandung beberapa informasi mengenai hasil dari kebanyakan hasil eksekusi intruksi aritmatik. Informasi ini dapat digunakan untuk altering arus program sebagai kegunaan untuk meningkatkan performa pengoperasian. Perlu diketahui bahwa register ini di-update setelah semua operasi ALU ( Arithmetic Logic Unit ). Hal tersebut seperti yang telah tertulis dalam datasheet khususnya pada bagian Intruction Set



Reference. Dalam hal ini untuk beberapa kasus dapat membuang kebutuhan penggunaan instruksi perbandingan yang telah didedikasikan serta dapat menghasilkan peningkatan dalam hal kecepatan dan kode yang lebih sederhana dan singkat. Register ini tidak secara otomatis tersimpan ketika memasuki sebuah rutin interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Namun hal ini harus dilakukan melalui software.

9. Bit 7

Merupakan bit Global Interrupt Enable. Bit ini harus di-set supaya semua perintah interupsi dapat dijalankan. Untuk fungsi interupsi individual akan dijelaskan pada bagian yang lain. Jika bit ini di-reset, maka semua perintah interupsi baik yang secara individual maupun yang secara umum akan diabaikan. Bit ini akan dibersihkan atau cleared oleh hardware setelah sebuah interupsi dijalankan dan akan di-set kembali oleh perintah RETI. Bit ini juga dapat di-set dan di-reset melalui aplikasi dengan instruksi SEI dan CLI.

10. Bit 6

Merupakan bit Copy Storage. Instruksi bit Copy Instruction BLD ( Bit Load ) dan BST ( Bit Store ) menggunakan bit ini sebagai asal atau tujuan untuk bit yang telah dioperasikan. Sebuah bit dari sebuah register dan Register File dapat disalin ke dalam bit ini dengan menggunakan instruksi BST, dan sebuah bit di dalam bit ini dapat disalin ke dalam sebuah bit di register pada Register File dengan menggunakan perintah BLD.

11. Bit 5

Merupakan bit Half Carry Flag. Bit ini menandakan sebuah Half Carry dalam beberapa operasi aritmatika. Bit ini berfungsi dalam aritmatik BCD.

12. Bit 4

Merupakan Sign bit. Bit ini selalu merupakan sebuah eksklusif di antara Negative Flag (N) dan Two's Complement Overflow Flag (V).

13. Bit 3

Merupakan bit Two's Complement Overflow Flag. Bit ini menyediakan fungsi aritmatika dua komplemen.

14. Bit 2

Merupakan bit Negative Flag. Bit ini menyediakan sebuah hasil negative di dalam sebuah fungsi logika atau aritmatika.

15. Bit 1

Merupakan bit Zero Flag. Bit ini mengindikasikan sebuah hasil nol " 0 " dalam sebuah fungsi aritmatika atau logika.

16. Bit 0

Merupakan bit Carry Flag. Bit ini mengindikasikan sebuah Carry atau sisa dalam sebuah fungsi aritmatika atau logika.

## 2.4 *Coin Acceptor*

*Coin Acceptor* adalah sebuah perangkat sensor koin yang biasa digunakan pada telepon umum atau *vending machine*. Pendeteksian dilakukan dengan memanfaatkan sensor logam yang cara kerjanya adalah dengan mengukur frekuensi resonansi dari detektor logam tersebut. Koin yang dimasukkan akan melewati sebuah koil pendeteksi, frekuensi keluaran osilator akan berbeda tergantung dari jenis koinnya (Irman, 2013).

Terdapat dua jenis *coin acceptor* yaitu *single coin* dan *multi coin detector*. Pada tugas akhir ini *Coin acceptor* yang digunakan adalah berjenis *multi coin detector*, yaitu sensor dapat mendeteksi lebih dari satu jenis koin tertentu. Pada *coin acceptor multi coin* dapat membaca lebihsatu jenis koin yang akan dijadikan referensi sebagai pembanding koin, dengan membandingkan antara frekuensi koin referensi dan koin yang dimasukkan, maka akan dapat dilakukan keputusan untuk menerima koin yang masuk atau menolak koin yang masuk. Artinya *coin acceptor* jenis ini dapat menerima lebih satu jenis koin, namun koin yang terbaca hanya koin yang sama dengan koin yang telah dijadikan referensi. Apabila ingin menambah jenis dapat dilakukan kalibrasi pada jenis koin yang dibutuhkan, maksimal penyimpanan pembacaan pada *Coin Acceptor* jenis ini adalah 3 jenis koin (Irman, 2013).

Jika koin yang dimasukkan mempunyai frekuensi yang sama, maka koin akan diterima oleh *coin acceptor* dan keluar melalui sisi bawah *coin acceptor*, pada sisi ini dapat ditambahkan wadah untuk pengumpulan koin yang telah diterima dan ketika koin diterima maka *coin acceptor* juga akan memberi sinyal bahwa koin yang telah dimasukkan tersebut sesuai dengan koin referensi. Sebaliknya, jika koin yang masuk frekuensinya tidak sama dengan koin referensi

maka koin tersebut akan ditolak dan dikeluarkan kembali pada tempat dimana koin tersebut dapat diambil kembali. Bentuk modul dari *Coin Acceptor* dapat dilihat pada Gambar 2.3 (Irman, 2013).

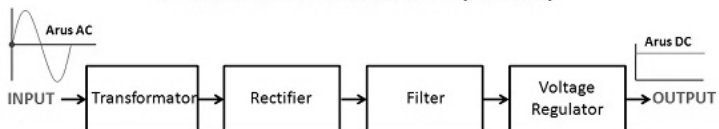


**Gambar 2. 3** *Coin Acceptor*(Irman, 2013).

## 2.5 Power Supply

*Power Supply* atau pencatu daya merupakan rangkaian elektronika yang dapat menghasilkan energi listrik atau sebagai sumber energi untuk rangkaian elektronika lainnya. Sumber arus dari *power supply* adalah arus bolak – balik (AC) dari pembangkit listrik yang kemudian diubah menjadi arus searah (DC). Untuk dapat melakukan hal tersebut *power supply* memerlukan perangkat yang bisa mengubah arus AC menjadi DC. Sebuah DC *power supply* pada dasarnya memiliki 4 bagian utama agar dapat menghasilkan arus DC yang stabil (Irman., 2013).

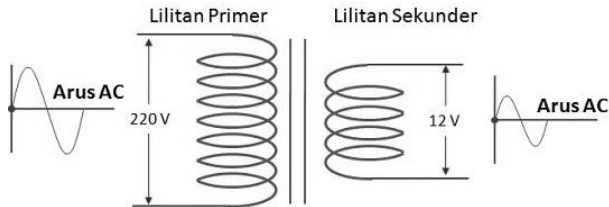
Keempat bagian utama tersebut diantaranya adalah *Transformer*, *rectifier*, *filter* dan *voltage regulator*. Berikut adalah diagram blok *power supply* pada Gambar 2.4.



**Gambar 2. 4** Diagram Blok *Power Supply* (Irman., 2013).

Gambar 2.5 adalah penjelasan singkat tentang prinsip kerja DC *power supply* pada masing-masing blok berdasarkan diagram blok diatas.

a. Transformator (Trafo)

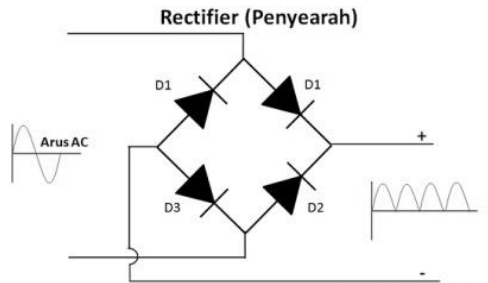


**Gambar 2. 5** Trafo *Step Down* (Irman., 2013).

Transformator yang digunakan untuk DC *power supply* adalah Transformer jenis *Step-down* yang berfungsi untuk menurunkan tegangan listrik sesuai dengan kebutuhan komponen Elektronika yang terdapat pada rangkaian adaptor (DC *power supply*). Transformator bekerja berdasarkan prinsip Induksi elektromagnetik yang terdiri dari 2 bagian utama yang berbentuk lilitan yaitu lilitan primer dan lilitan sekunder. Lilitan primer merupakan *input* dari transformator sedangkan *output*-nya pada lilitan sekunder. Meskipun tegangan telah diturunkan, *output* dari transformator masih berbentuk arus bolak-balik (arus AC) yang harus diproses selanjutnya (Irman., 2013).

b. *Rectifier* (Penyearah Gelombang)

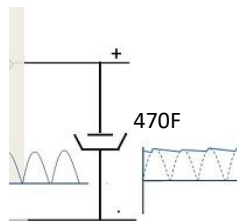
*Rectifier* atau penyearah gelombang adalah rangkaian elektronika dalam *power supply* yang berfungsi untuk mengubah gelombang AC menjadi gelombang DC setelah tegangannya diturunkan oleh transformator *step down*. Rangkaian *Rectifier* biasanya terdiri dari komponen dioda. Terdapat 2 jenis rangkaian *Rectifier* dalam *power supply* yaitu “*Half Wave Rectifier*” yang hanya terdiri dari 1 komponen dioda dan “*Full Wave Rectifier*” yang terdiri dari 2 atau 4 komponen diode pada Gambar 2.6 (Irman., 2013).



**Gambar 2. 6** Rectifier pada Power Supply (Agfianto, 2002).

c. *Filter (Penyaring)*

Dalam rangkaian *power supply* (Adaptor), *filter* digunakan untuk meratakan sinyal arus yang keluar dari *Rectifier*. *Filter* ini biasanya terdiri dari komponen kapasitor (kondensator) yang berjenis elektrolit atau ELCO (*Electrolyte Capacitor*) pada Gambar 2.7.



**Gambar 2. 7** Filter pada Power Supply (Agfianto, 2002).

d. *Pengatur Tegangan (Voltage Regulator)*

Untuk menghasilkan tegangan dan arus DC (arus searah) yang tetap dan stabil, diperlukan *voltage regulator* yang berfungsi untuk mengatur tegangan sehingga tegangan *output* tidak dipengaruhi oleh suhu, arus beban dan juga tegangan *input* yang berasal *output filter*. *Voltage regulator* pada umumnya terdiri dari Dioda Zener, Transistor atau IC (*Integrated Circuit*). Pada DC *power supply* yang canggih, biasanya *voltage regulator* juga dilengkapi dengan *short circuit protection* (perlindungan atas hubung singkat), *current limiting* (pembatas Arus) ataupun *over voltage protection* (perlindungan atas kelebihan tegangan) (Latif, 2012).

## 2.6 RTC (*Real Time Clock*)

RTC adalah sebuah rangkaian elektronika yang berfungsi sebagai acuan waktu. Pada umumnya RTC digunakan pada alat elektronika yang membutuhkan akurasi waktu yang sesuai dengan waktu dunia. RTC berbeda dengan jam biasa karena RTC umumnya hanya dalam bentuk IC. Dalam penggunaannya, dengan adanya RTC sebuah sistem dapat fokus dengan tugas utamanya. Selain itu, RTC mempunyai sumber daya yang berbeda dari sistem. Sehingga ketika sistem dimatikan RTC masih berfungsi dan waktunya tidak akan berhenti atau *ter – reset* saat *restart*. Data – data yang tersimpan pada IC DS3207 disimpan pada *register* 00H untuk detik, 01H untuk menit, 02H untuk jam, 03H untuk hari, 04H untuk tanggal, 05H untuk bulan 06H untuk tahun, 07H untuk kontrol dan RAM 56x8 pada *register* 08H – 3FH. *Register* tersebut bisa diakses oleh mikrokontroler melalui bus I2C (Agfianto, 2002).

Pada pin 1 dan pin 2 dihubungkan dengan kristal eksternal 32,768 KHz. Pin 3 Vbat di hubungkan dengan baterai eksternal 3V. Pin 4 GND dapat kita hubungkan dengan *ground* pada baterai eksternal. Pin 8 Vcc di hubungkan ke Vcc minimum sistem. Pada pin 5, 6, dan 7 adalah sinyal data dari RTC ke mikrokontroler, di antaranya pin 7 adalah SQM, pin 6 adalah SCL dan pin 5 adalah SDA dan di masing-masing sinyal data tersebut kita berikan resistor 1k $\Omega$  hingga 10k $\Omega$  sebagai *pull up*, kecuali pin 7 yaitu SQM. Untuk lebih jelasnya dapat dilihat pada Gambar 2.8 (Agfianto, 2002).



**Gambar 2. 8** RTC DS1307 (Agfianto, 2002).

## 2.7 *Liquid Cristal Display (LCD) 20x4*

LCD (*Liquid Cristal Display*) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD yang dipakai merupakan tipe berkarakter 20x4 baris, yang dapat menampilkan 20 karakter dengan 4 baris.

Pada aplikasinya tidak semua pin pada LCD 20x4 terpakai, melainkan hanya pin-pin yang dihubungkan dengan mikrokontroler yang digunakan saja. Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. *Interface* LCD merupakan sebuah paralel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 *nibble* data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa *clock* EN setiap *nibble*-nya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroler mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus men-*set* EN ke kondisi *high* “1” dan kemudian mengatur dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus (Heri, 2015).

Saat jalur lainnya sudah siap, EN harus diatur ke “0” dan tunggu beberapa saat (tergantung pada *datasheet* LCD), dan *set* EN kembali ke *high* “1”. Ketika jalur RS berada dalam kondisi *low* “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi *high* atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan di layar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus di-*set* ke “1”. Jalur kontrol R/W harus berada dalam kondisi *low* (0) saat informasi pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi *high* “1”, maka program akan melakukan *query* (pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu *Get LCD* status (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu di-*set* ke “0”. Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan

DB7. Mengirim data secara paralel baik 4-bit atau 8-bit merupakan 2 mode operasi primer. (Agfianto, 2002).

Mode 8-bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 pin I/O (3 pin untuk kontrol, 8 pin untuk data). Sedangkan mode 4 bit minimal hanya membutuhkan 7-bit (3 pin untuk kontrol, 4 pin untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroler dan LCD. Jika bit ini di-*set* ( $RS = 1$ ), maka *byte* pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini di-*reset* ( $RS = 0$ ), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca. Bentuk dari LCD dapat dilihat pada Gambar 2.9 (Agfianto, 2002).



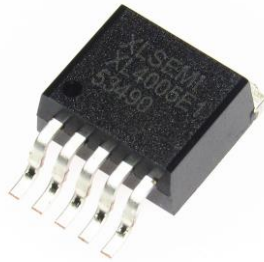
**Gambar 2. 9** LCD 20x4 (Agfianto, 2002).

## **2.8 XL4005**

XL4005 adalah konverter DC / DC frekuensi tetap 300MHz (step-down) DC, yang mampu menggerakkan beban 5A dengan efisiensi tinggi, riak rendah dan garis yang sangat baik serta pengaturan beban. Dengan mewajibkan jumlah minimum komponen eksternal, pengatur mudah digunakan dan termasuk kompensasi frekuensi internal dan osilator frekuensi tetap. Rangkaian kontrol PWM dapat menyesuaikan rasio tugas secara linear dari 0 hingga 100%. Fungsi pengaktifan, fungsi perlindungan lebih dari saat ini dibangun di dalam. Ketika fungsi perlindungan pendek terjadi, frekuensi operasi akan berkurang dari 300KHz menjadi 60KHz. Blok kompensasi internal dibangun untuk meminimalkan jumlah komponen eksternal. XL 4005 mempunyai Rentang tegangan input 5V -



32V, output tegangan dapat diatur 0.8V -30V, duty cycle maksimal 100%, drop out minimal 0.6V, frekuensi switching 300 Khz tetap, Kemampuan keluaran arus konstan 5A, optimalkan daya MOSFET internal, pengaturan jalur dan beban yang sangat baik, kemampuan shutdown TTL, EN pin dengan fungsi hysteresis, Dibangun dalam fungsi thermal shutdown, Dibangun pada output fungsi perlindungan singkat, tersedia dalam paket TO-263. Untuk lebih jelasnya dapat dilihat pada Gambar 2.10 (Latif, 2012).



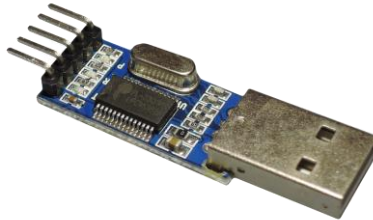
**Gambar 2. 10** XL4005 (Lingga, 2006)

## **2.9 USB to TTL Serial**

USB to TTL adalah suatu standar komunikasi serial transmisi data antar dua peralatan elektronik. USB to TTL dibuat pada tahun 1962 oleh Electronic Industry Association and Telecommunication Industry Association dan ada dua hal pokok yang diatur oleh USB to TTL, yaitu: bentuk signal dan level tegangan yang dipakai. Selain digunakan pada peralatan PLC, sebenarnya sistem USB to TTL ini sering berhubungan dengan kita pada kehidupan sehari-hari, antara lain: Komunikasi PC komputer dengan *mouse*, *keyboard* atau *scanner*. Satu hal yang jelas adalah USB to TTL ini akan diaplikasikan pada semua sistem peralatan yang berbasis computer atau mikro-kontroler (Lingga, 2006).

Komunikasi data secara serial dilakukan dengan metode pengiriman data secara bit per bit atau satu per satu secara berurutan dan itu berbeda dengan sistem paralel yang mengirim data secara serentak. kecepatan transfer data USB to TTL cukup rendah, kecepatan maksimal hanya 19200 bits/sekon. Pengiriman data bisa dilakukan secara satu arah atau dua arah. Jika Anda hanya membu-

tuhkan komunikasi satu arah maka Anda cukup menggunakan dua kabel yaitu kabel “TX” sebagai pengirim data dan kabel “Rx” sebagai penerima data. Sedangkan, untuk membuat sistem komunikasi dua arah maka kabel yang Anda butuhkan adalah 3 unit kabel, yaitu kabel Tx, Rx dan GND (*ground*). Bentuk dari USB to TTL Serial dapat dilihat pada Gambar 2.11 (Lingga, 2006).



**Gambar 2. 11** Modul USB to TTL Serial (Latif, 2012)

## **2.10 CodeVisionAVR**

Ada banyak jenis software yang dapat digunakan sebagai editor yang sekaligus menyediakan compiler untuk mikrokontroler Atmel AVR dengan menggunakan bahasa C, diantaranya MikroC for AVR, WinAVR, Image Craft ICC AVR, IAR Embedded Workbench for AVR, dan CodeVision AVR. CodeVisionAVR menyediakan sebuah editor yang didesain untuk menghasilkan program C secara otomatis untuk mikrokontroler AVR. Program C yang akan diimplementasikan menggunakan standar ANSI C yang sesuai dengan arsitektur AVR. CodeVisionAVR adalah sebuah compiler C yang telah dilengkapi dengan fasilitas Integrated Development Environment (IDE) dan didesain agar dapat menghasilkan kode program secara otomatis untuk mikrokontroler Atmel AVR. Program ini dapat berjalan dengan menggunakan sistem operasi Windows® XP, Vista, Windows 7, dan Windows 8, 32-bit dan 64-bit. Integrated Development Environment (IDE) telah dilengkapi dengan fasilitas pemrograman chip melalui metode In-System Programming sehingga dapat secara otomatis mentransfer file program ke dalam chip mikrokontroler AVR setelah sukses dikompilasi. Software In-System Programmer didesain untuk bekerja ketika dihubungkan dengan development board STK500, STK600, AVRISP mkII, AVR Dragon,

AVRProg (AVR910 application note), Atmel JTAGICE mkII, Kanda System STK200+STK300, Dontronics DT006, Vogel Elektronik VTEC-SIP, Futurlec JRAVR and MicroTronics ATCPU, dan Mega2000 (Heri, 2015).

Untuk meningkatkan kehandalan program, maka pada CodeVisionAVR juga terdapat kumpulan pustaka (library) untuk:

1. Modul LCD Alphanumeric.
2. Philips I2C bus.
3. National Semiconductor Sensor Temperatur LM75.
4. Philips PCF8563, PCF8583, dan Maxim/Dallas Semiconductor Real Time Clock DS1302 dan DS1307.
5. Maxim/Dallas Semiconductor 1 wire protocol.
6. Maxim/Dallas Semiconductor Sensor Temperatur DS1820, DS18S20, dan DS18B20.
7. Maxim/Dallas Semiconductor Termometer/Thermostat DS1621.
8. Maxim/Dallas Semiconductor EEPROMs DS2430 dan DS2433.
9. SPI.
10. Power Management.
11. Delays.
12. Gray Code Conversion.
13. MMC/SD/SD HC Flash memory cards low level access.
14. Akses FAT pada MMC/SD/SD HC Flash memory card.

CodeVisionAVR dapat menghasilkan kode program secara otomatis melalui fasilitas CodeWizardAVR Automatic Program Generator. Dengan adanya fasilitas ini maka penulisan program dapat dilakukan dengan cepat dan lebih efisien. Seluruh kode dapat diimplementasikan dengan fungsi sebagai berikut:

1. Identifikasi sumber reset
2. Mengatur akses memori eksternal
3. Inisialisasi port input/output
4. Inisialisasi interupsi eksternal
5. Inisialisasi timer/counter dan watchdog timer
6. Inisialisasi USART dan interupsi buffer untuk komunikasi serial
7. Inisialisasi komparator analog dan ADC
8. Inisialisasi interface SPI dan two wire interface (TWI)
9. Inisialisasi interface CAN

10. Inisialisasi I2C Bus, sensor suhu LM75, thermometer/thermostat DS1621, dan real time clock PCF8563, PCF8583, DS1302, DS1307
11. Inisialisasi 1 wire bus dan sensor suhu DS1820/DS18S20
12. Inisialisasi modul LCD

## 2.11 Bahasa C

Bahasa Pemrograman C diciptakan dan dikembangkan oleh Brian Kernighan dan Denis Ritchie di Bell Research Labs. Bahasa Pemrograman C secara khusus diciptakan dengan tujuan agar para programmer (orang yang membuat program komputer) dapat mengakses seluruh internal register. I/O slots dan absolute address dari sebuah komputer. Pada awal tahun 1960-an, sistem operasi komputer mulai menjadi jauh lebih kompleks dari sebelumnya karena adanya pengenalan multi terminal dan kemampuan multi prosesor. Pada saat itu, sistem operasi diciptakan dengan menggunakan bahasa assembly (bahasa pemrograman tingkat rendah). Dimana banyak pengembang yang menyadari bahwa suatu sistem operasi dapat dikembangkan lebih lagi, tidak cukup hanya dengan menggunakan bahasa assembly. Inilah asal mula Bahasa C yang diimplementasikan pada Digital Equipment Corporation PDP-7. Pada perkembangan selanjutnya Bahasa Pemrograman C digunakan untuk mengimplementasikan Sistem Operasi Unix (Heri, 2015).

Bahasa C atau C++ adalah suatu bahasa pemrograman. Bahasa C termasuk sebagai bahasa pemrograman tingkat menengah, maksudnya bahasa C bisadipelajari dengan lebih mudah karena mudah dimengerti tetapi mempunyai kemampuan yang tinggi. Bahasa C bisa digunakan untuk merencanakan program untuk segala kebutuhan, baik untuk aplikasi bisnis, matematis atau bahkan game

Bahasa Pemrograman C telah mengalami banyak evolusi sejak awal diciptakanya hingga saat ini. Saat ini, banyak Bahasa Pemrograman yang merupakan turunan/varian/keluarga dari Bahasa C seperti: C ++, Java Script, PHP, Java, perl dan lain sebagainya. Bisa juga dibilang bahasa C adalah induk dari bahasa pemrograman saat ini. Beberapa kelebihan dari bahasa C adalah sebagai berikut :

1. Banyak memiliki operator untuk mengolah / memanipulasi data.
2. Bahasa C termasuk sebagai bahasa yang terstruktur sehingga program dapat lebih mudah dipahami atau dikembangkan.

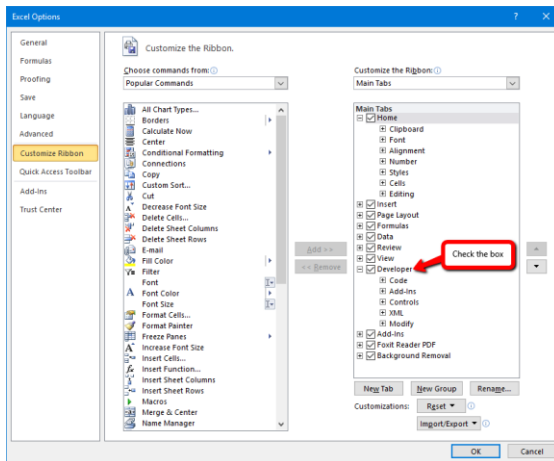
3. Bahasa C lebih mudah dimengerti karena lebih mirip kepada bahasa manusia.
4. Menenal data pointer.
5. Bahasa C tersedia hampir di semua jenis computer.
6. Kode bahasa C sifatnya adalah portable dan fleksibel untuk semua jenis computer.
7. Bahasa C hanya menyediakan sedikit kata-kata kunci, hanya terdapat 32 kata kunci.
8. Proses executable program bahasa C lebih cepat.
9. Dukungan pustaka yang banyak.
10. C adalah bahasa yang terstruktur.
11. Bahasa C termasuk bahasa tingkat menengah

## 2.12 PLX-DAQ

PLX-DAQ adalah pengaya sederhana untuk Microsoft Excel yang membuatnya mudah digunakan untuk membuat lembar log excel dalam keperluan laboratorium atau tujuan DAQ lainnya. Penggunaannya dapat untuk mengumpulkan data sensor dan kemudian menggunakan Excel untuk menggambar diagram atau diagram untuk eksperimen yang diinginkan, sehingga PLX-DAQ dan Mikrokontroler sangat mudah digunakan. Langkah – langkah penggunaan PLX-DAQ adalah sebagai berikut :

1. *Installing PLX DAQ*  
Pertama perlu mengunduh file ZIP dari situs webnya. Setelah itu un-zip dan kemudian masukkan pengaturan dan ikuti proses instalasi selanjutnya akan menemukan bahwa ada folder desktop yang disebut "PLX-DAQ". Masukkan dan buka file Excel.
2. *Mengaktifkan Macro and Active-X*  
Seperti yang diketahui bahwa *Macro* adalah program kecil yang mengotomatiskan langkah yang digunakan untuk meningkatkan produktivitas, namun jangan mencoba menjalankan *Macro* apa pun yang tidak diketahui sumbernya, karena mungkin virus yang dapat menghapus *file*. Pada Excel tidak mengizinkan *Macro* untuk dijalankan secara default. Untuk mengaktifkannya, Anda harus melakukan hal-hal berikut :
  1. Show the hidden (Developer tab).
  2. Enable the Macros.

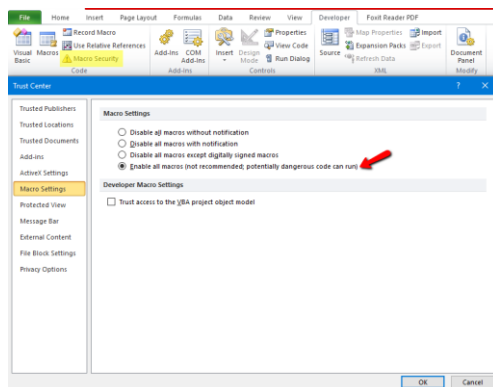
## 1. Menampilkan “Developer Tab”



**Gambar 2. 12** Menampilkan “Developer Tab”

1. Klik File tab.
2. Klik Options.
3. Klik Customize Ribbon.
4. Dibawah Customize the Ribbon dan dibawah Main Tabs, pilih centang pada Developer.

## 2. Mengaktifkan Macro



**Gambar 2. 13** Mengaktifkan Makro

### 3. Contoh Program pada Arduino

PLX-DAQ bergantung pada komunikasi serial UART sehingga dapat menggunakan Serial.print () dan fungsi lain dari serial. Berikut adalah program sederhana yang mengirimkan dua data :

1. Waktu dalam milidetik terbentuk saat Arduino diatur ulang
2. Tegangan pada input analog A0.

```
unsigned long int milli_time;    //variable to hold the time

float voltage;                  //variable to hold the voltage form A0

void setup() {
    Serial.begin(128000);        //Fastest baudrate
    Serial.println("CLEARDATA"); //This string is defined as a
                                // command for the Excel VBA
                                // to clear all the rows and
    columns
    Serial.println("LABEL,Computer Time,Time (Milli Sec.),Volt");
                                //LABEL command creates label
    for
                                // columns in the first row
    with bold font
}

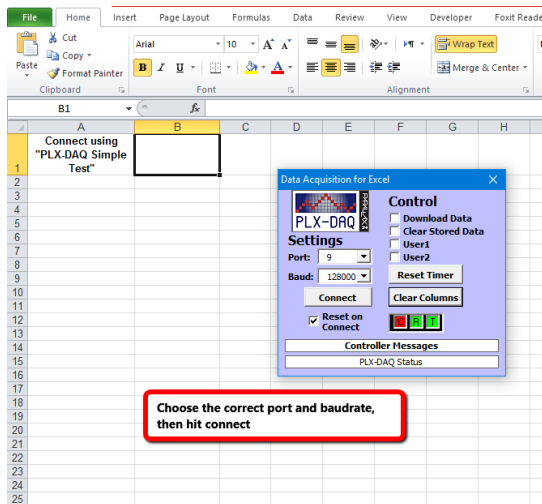
void loop() {
    milli_time = millis();
    voltage = 5.0 * analogRead(A0) / 1024.0;
    Serial.print("DATA,TIME,");
    Serial.print(milli_time);
    Serial.print(",");
    Serial.println(voltage);

    delay(100);                 //Take samples every one second
}
```

**Gambar 2. 14** Contoh Program Arduino

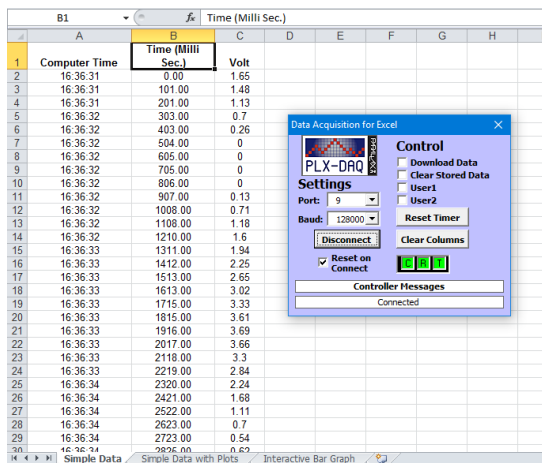
### 4. Menjalankan PLX-DAQ

Pada Excel awal membuka file "PLX-DAQ" lalu secara otomatis Excel memberitahukan bahwa ada makro Active-X yang ingin dijalankan, klik OK, maka tampilan seperti pada Gambar 2.15 akan terjadi.



**Gambar 2. 15** Konfigurasi PLX – DAQ

Pilihlah port dan baud rate yang benar, dalam kasus tersebut “COM 9” dan “128000”, lalu tekan “Connect”.



**Gambar 2. 16** PIX-DAQ Berhasil Mengirim Data



### 2.13 Khazama

Khazama AVR programmer merupakan salah satu software untuk menulis (mendownload) file. hex ke board mikrokontroler. Tujuan dari program ini adalah ukuran yang ringan, program cepat, handal dan mudah digunakan. Langkah – langkah penggunaan Khazama :

1. Buka program khazama
2. pilih *read chip signature* ( berfungsi membaca *chip/* dapat difungsikan)



**Gambar 2. 17** Read chip signature

3. Muncul chip signature (menandakan mikrokontroler bisa difungsikan)

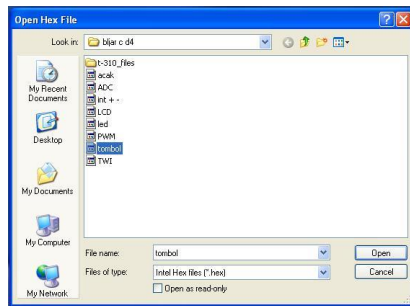


**Gambar 2. 18** Pemberitahuan *Chip Signature*

4. Pilih file> *load flash* ( memasukan file.hex)



**Gambar 2. 19** Memasukkan File.hex



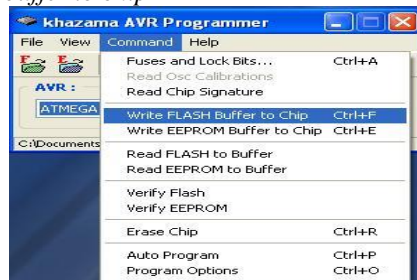
**Gambar 2. 20** Pilih File

5. Pilih Auto Program ( muncul seperti gambar dibawah, menandakan file.hex berhasil di download)



**Gambar 2. 21** File.hex berhasil di download

Ataupun bisa juga memilih manual, pilih *Command* > klik *write flash buffer to chip*



**Gambar 2. 22** Memilih *Manual*

Gambar dibawah, menunjukan program berhasil di download



**Gambar 2. 23** Program Berhasil di *Download*

-----Halaman ini sengaja dikosongkan-----

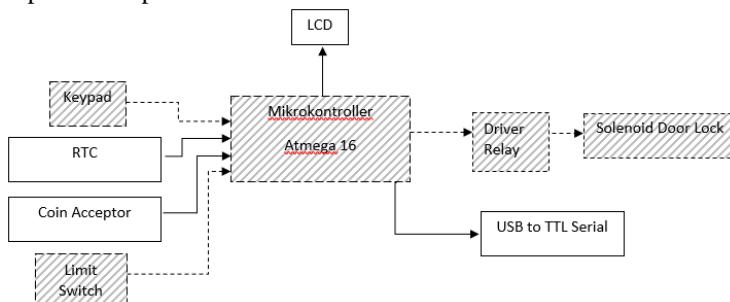
## BAB III

### PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini akan membahas mengenai tahapan yang dilakukan terhadap perancangan dan pembuatan tugas akhir rancang bangun alat penghitung pendapatan koin pada loker susun berbasis mikrokontroler. Blok diagram alat Tugas Akhir pada Gambar 3.1 merupakan sistem kerja Tugas Akhir rancang bangun alat penghitung pendapatan koin pada loker susun berbasis mikrokontroler. Sistem kerja secara keseluruhan dikontrol oleh mikrokontroler ATmega 16 yang terintegrasi dengan rangkaian dan komponen – komponen pendukung. Adapun fungsi masing – masing komponen yang terdapat pada diagram perancangan alat Tugas Akhir, antara lain :

#### 3.1 Blok Fungsional Sistem

Sebelum melakukan perancangan perangkat keras dan perangkat lunak, diperlukan sebuah perancangan blok fungsional sistem berupa blok diagram yang menjelaskan sistem kerja secara keseluruhan Tugas Akhir ini, untuk bagian yang diarsir dikerjakan oleh rekan kelompok. Secara keseluruhan blok fungsional sistem dapat dilihat pada Gambar 3.1.



**Gambar 3. 1** Blok Fungsional Sistem

Sesuai dengan gambar blok fungsional sistem diatas Mikrokontroler Atmega 16 digunakan sebagai kontroler untuk mengatur data yang muncul pada LCD 16x4 yang digunakan sebagai *interface* bagi pengguna, dan juga Mikrokontroler Atmega 16

mengontrol masukan dari *Coin Acceptor* agar dapat menerima masukan koin dari pengguna. Selain itu juga Mikrokontroler Atmega 16 juga mengontrol RTC sehingga dapat mengatur waktu secara *real time*. Dan juga menggunakan komunikasi USB to TTL digunakan untuk melihat data penggunaan loker ke dalam PC. Fungsi dari masing – masing komponen juga dapat dilihat dibawah ini :

1. *Coin Acceptor*

Fungsi dari *Coin Acceptor* sebagai tempat pembayaran loker dengan menggunakan uang koin sebesar Rp 500,-.

2. LCD 20x4

Berfungsi sebagai *interface password* dan perintah awal dari sistem.

3. *Real Time Clock*

Merupakan modul yang berfungsi untuk menyediakan data waktu dan tanggal secara *Real Time*.

4. USB to TTL Serial

Sebagai komunikasi serial antara Mikrokontroler Atmega 16 dengan komputer.

### 3.2 Cara Kerja Alat

Cara kerja dari alat ini adalah mula – mula setelah password telah di set maka RTC akan mulai menghitung. Penggunaan RTC digunakan untuk melakukan penyimpanan lama waktu sewa dari loker yang digunakan. RTC akan mulai menghitung waktu lama sewa pada saat loker digunakan. Data pada RTC akan digunakan untuk mendapatkan nilai waktu dari loker digunakan. Selanjutnya data lama waktu RTC digunakan untuk melakukan perhitungan dari tarif yang digunakan. RTC akan mulai menghitung waktu lama sewa pada saat loker digunakan. Data pada RTC akan digunakan untuk mendapatkan nilai tarif pada loker yang digunakan. Pada tahap selanjutnya dilakukan sistem pembayaran menggunakan *Coin* sebesar Rp 500 dan pada alat ini dapat membaca dua tipe *Coin* Rp 500 yaitu perak dan kuning. *Coin* akan dibayarkan sesuai dengan tarif yang telah ditentukan dari lama waktu penggunaan loker sebelumnya. Setelah selesai terbayar maka selanjutnya akan mengirimkan serial data menggunakan komunikasi serial. Komunikasi serial digunakan untuk menampilkan tarif penggunaan loker dari loker yang telah

digunakan ke Komputer. Data dikirim setelah pembayaran loker telah selesai dilakukan. Data tersebut akan ditampilkan pada Microsoft Excel berupa tabel waktu dan tarif penggunaan.

### **3.3 Perancangan *Hardware***

Perancangan *Hardware* dilakukan dengan merancang rangkaian elektronika dan rancangan mekanik. Perancangan *hardware* meliputi :

1. Rangkaian *Coin Acceptor*
2. Rangkaian *Real Time Clock DS1307*
3. Rangkaian USB to TTL Serial
4. Rangkaian LCD 20x4

#### **3.3.1 *Setting Coin Acceptor* CH-923**

Pada alat Tugas Akhir ini dilengkapi dengan sistem pembayaran menggunakan *Coin*. Modul yang digunakan untuk mendeteksi *Coin* tersebut adalah modul *Coin Acceptor*. *Coin Acceptor* bekerja mengenali jenis *Coin* berdasarkan pada bahan, berat dan diameter dari *Coin* tersebut. Pada Tugas Akhir ini pembayaran dilakukan dengan menggunakan *Coin* yang berupa uang logam Rp 500 perak dan Rp 500 kuning. Pin yang digunakan pada modul *Coin Acceptor* CH-923 adalah pin VCC, pin GND, dan pin *Coin*. Modul *Coin Acceptor* menggunakan tegangan sebesar +12 V yang terhubung ke pin VCC dan GND pada *Power Supply*. Sedangkan pin *Coin* digunakan sebagai *input* pulsa ke Mikrokontroler ATmega16. Untuk menggunakan *Coin Acceptor* maka terlebih dahulu dilakukan setting untuk menentukan jenis *Coin* yang dapat digunakan pada alat ini.

Cara *setting* dari *Coin Acceptor* CH-923 sebagai berikut :

1. Hubungkan *Coin Acceptor* CH-923 dengan Vcc sebesar 12V kemudian akan terdengar bunyi “beep” dan LED menyala.
2. Tekan tombol “Add” dan “Minus” secara bersamaan selama kurang lebih 3 detik kemudian akan tampil huruf “A” pada display seven segment.
3. Tekan tombol “SET” satu kali, tombol “E” akan tampil pada display, selanjutnya tentukan jumlah jenis *Coin* dengan menekan tombol “Add” atau “Minus” kemudian tekan kembali tombol “SET”.

4. Pada display akan tampil huruf "H1" kemudian gunakan tombol "Add" dan "Minus" untuk mengatur berapa kali sampling akan dilakukan dengan menggunakan jenis *Coin* pertama, jumlah sampling sebaiknya berkisar 15-20 sampling. Selanjutnya tekan kembali tombol "SET".
5. Pada display akan tampil huruf "P1" kemudian gunakan tombol "Add" dan "Minus" untuk memilih jumlah pulsa yang ingin dikeluarkan pada pin Sinyal keluaran pada *Coin* yang digunakan. Jumlah pulsa sebaiknya tidak sama untuk tiap jenis *Coin*. Selanjutnya tekan tombol "SET".
6. Pada display akan tampil huruf "F1" kemudian gunakan tombol "Add" dan "Minus" untuk mengatur akurasi. Nilai akurasi berkisar antara 1 – 30 dimana nilai 1 adalah untuk pendeteksian paling akurat. Pada praktek dilakukan nilai 7 – 10 sudah cukup akurat untuk digunakan. Selanjutnya tekan tombol "SET".
7. Sudah selesai untuk *setting* untuk jenis *Coin* pertama, jika *Coin* yang digunakan lebih dari satu maka akan diulangi lagi pada langkah ke 4 -6 sampai sesuai dengan jumlah *setting Coin*.
8. Tekan kembali tombol "SET" maka akan tampil huruf "E" pada display, selanjutnya matikan catu daya. Pada saat ini *setting* yang dilakukan tadi telah tersimpan.  
Setelah dilakukan *setting* juga dilakukan *sampling* untuk mengerjakan cara nomor 4 pada *setting*. Cara melakukan *sampling* adalah sebagai berikut :
  1. Menyalakan catu daya.
  2. Tekan tombol "SET" maka akan tampil huruf "A1" pada display.
  3. Mulai melakukan *sampling* untuk jenis *Coin* pertama dengan cara memasukkan *Coin* sesuai dengan jumlah *setting* yang dilakukan pada "H1". Setelah memasukkan *Coin* sesuai dengan jumlah *sampling* maka akan kembali tampil huruf "A1".
  4. Tekan tombol "SET" selanjutnya huruf "A2" akan tampil. Lakukan *sampling* lagi pada jenis *Coin* kedua dengan cara yang sudah dilakukan sebelumnya.
  5. Lakukan kembali langkah ke 4 sampai semua jenis *Coin* selesai melakukan *sampling*. Selanjutnya restart dengan

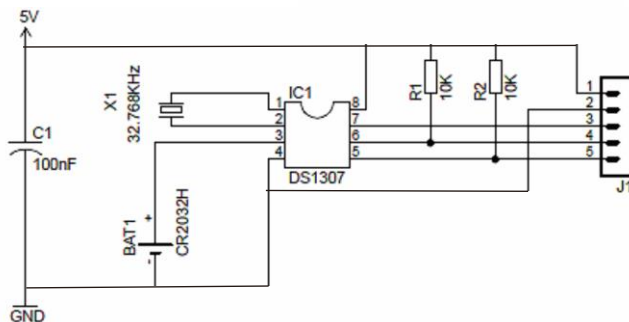


mencabut catu daya dan menancapkannya kembali maka akan terdengar bunyi “beep” dan *Coin Acceptor* CH-923 sudah siap digunakan.

Pada *setting* yang telah dilakukan jenis *Coin* yang digunakan adalah sebanyak 2 jenis sehingga pada saat display seven segment menampilkan huruf “E” maka *setting* dilakukan dengan cara menekan tombol “Add” atau “Minus” sampai bernilai 2. Untuk jumlah proses *sampling* yang dilakukan pada proses *sampling* setiap jenis *Coin* adalah sebanyak 20 kali sehingga pada saat display seven segment menampilkan “H1” dan “H2” dilakukan *setting* dengan menekan tombol “Add” atau “Minus” sampai menunjukkan angka 20. Selanjutnya untuk melakukan *setting* banyaknya pulsa yang dihasilkan oleh *Coin Acceptor* CH-923 ketika *Coin* terdeteksi adalah pada saat display seven segment menampilkan “P1” maka dilakukan *setting* terhadap pulsa keluaran *Coin Acceptor* dengan menggunakan tombol “Add” atau “Minus” untuk pada *setting* alat ini pulsa yang dipilih adalah 1 untuk *Coin* kuning dan pulsa 2 untuk *Coin* perak.

### 3.3.2 Perancangan RTC(*Real Time Clock*) DS1307

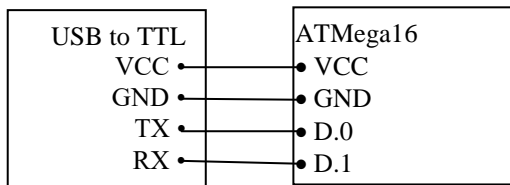
Pemakaian RTC dalam alat ini digunakan untuk menampilkan waktu pada LCD serta yang paling utama difungsikan sebagai waktu yang berpengaruh pada lama pemakaian dari loker yang digunakan sehingga RTC tersebut akan mempengaruhi harga dari lama penggunaan loker. Gambar 3.2 berikut adalah rangkaian modul RTC DS1307.



**Gambar 3. 2** Rangkaian RTC DS1307

### 3.3.3 Perancangan USB to TTL Serial

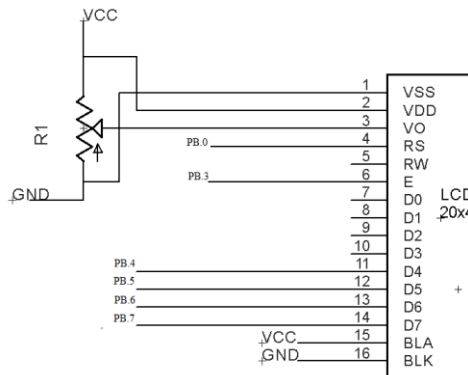
Pemakaian USB to TTL Serial dalam alat ini berfungsi sebagai serial komunikasi antara mikrokontroler Atmega 16 dan PC. Sehingga USB to TTL digunakan untuk mengirimkan data pada Atmega 16 yang berupa total pendapatan penyewaan loker tiap harinya yang dikirimkan ke PC. USB to TTL berfungsi sebagai serial komunikasi antara mikrokontroler Atmega 16 dan PC. Gambar 3.3 berikut adalah rangkaian modul yang terdapat pada USB to TTL :



**Gambar 3. 3** Rangkaian USB to TTL

### 3.3.4 Konfigurasi LCD dengan Mikrokonroler Atmega 16

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD yang digunakan adalah LCD dengan jumlah karakter 20x4, LCD berfungsi untuk menampilkan status kerja alat. Gambar 3.4 adalah skematik konfigurasi Mikrokontroler Atmega 16 dengan LCD 20x4 :



**Gambar 3. 4** Rangkaian LCD 20x4 dengan mikrokontroler

LCD (*Liquid Cristal Display*) disini berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD yang dipakai merupakan tipe berkarakter 20x4 baris, yang dapat menampilkan 20 karakter dengan 4 baris.

Pada aplikasinya tidak semua pin pada LCD 20x4 terpakai, melainkan hanya pin-pin yang dihubungkan dengan mikrokontroler yang digunakan saja. *Interface* dari LCD merupakan sebuah paralel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD.

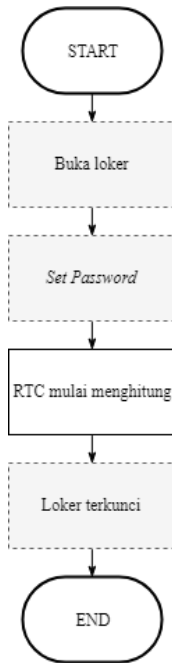
### **3.4 Perancangan Software**

Perancangan *Software* dilakukan setelah tahap perancangan elektronik telah dilakukan dan telah dirangkai. Perancangan *Software* dilakukan sehingga alat yang telah dirancang pada Tugas Akhir ini dapat mencapai tujuannya. Tanpa adanya perancangan *Software* maka rancangan alat dari Tugas Akhir tidak akan berjalan sesuai dengan tujuan.

Pada rancangan alat Tugas Akhir ini setiap komponen memiliki peranan sendiri dalam mencapai tujuannya. RTC berfungsi sebagai untuk menampilkan waktu dan perhitungan waktu lama penggunaan pada loker.

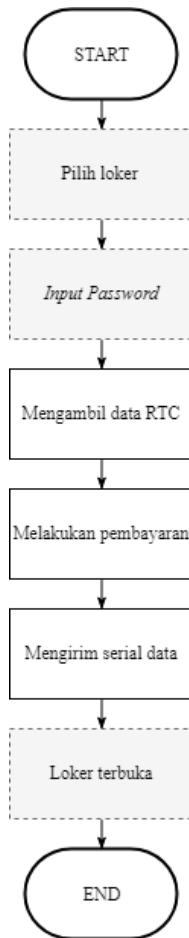
*Coin Acceptor* digunakan sebagai sarana untuk mendeteksi koin yang masuk dan dibaca oleh mikrokontroler. Selanjutnya untuk mengirim serial data menggunakan USB to TTL yang akan dikirimkan ke PC.

*Software* yang digunakan untuk alat Tugas Akhir ini adalah pemograman CodeVisionAVR. Bagian yang dikerjakan adalah yang berwarna putih sedangkan yang dikerjakan oleh rekan kelompok adalah berwarna abu – abu. Beberapa tahapan pemograman yang dilakukan untuk menjalankan alat penghitung pendapatan koin pada loker susun adalah pada Gambar 3.5.



**Gambar 3. 5** *Flowchart* proses awal menyewa loker

*Flowchart* dimulai dari start sebagai pertanda awal mula dari sistem. Diawal penggunaan yaitu pengguna sewa loker melakukan set password yang pada tahap ini pengerjaan *Software* dilakukan oleh rekan kelompok. Setelah set password telah berhasil dilakukan maka tahap selanjutnya adalah mengambil data waktu pada saat password telah set. Waktu tersebut diambil dari modul RTC yang yang berjalan, setelah itu RTC akan mulai menghitung lama penggunaan dari loker tersebut pada Gambar 3.6.



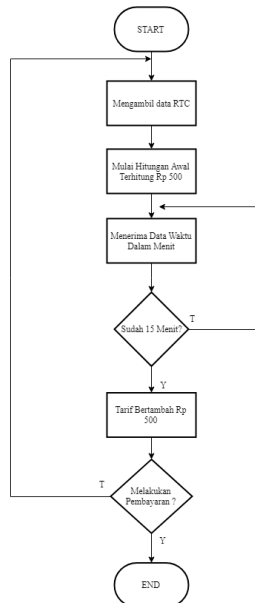
**Gambar 3. 6** Flowchart proses pengambilan barang pada loker

*Flowchat* dimulai dari *start* sebagai pertanda awal mula dari sistem. Sistem dimulai dari pengguna berhasil memasukkan *password* untuk membuka loker dari *password* yang telah diset sebelumnya namun pada tahap ini perancangan *Software* dikerjakan oleh rekan kelompok. Selanjutnya dilakukan pengambilan data RTC pada waktu pengambilan barang untuk mengambil nilai yang

digunakan sebagai data dari lama penggunaan waktu penggunaan loker. Data waktu lama penggunaan loker tersebut kemudian digunakan sebagai hitungan untuk menghasilkan nilai nominal harga yang harus dibayar oleh pengguna. Kemudian pembayaran dilakukan dengan memasukkan *Coin* bernilai Rp 500 perak atau kuning kedalam *Coin Acceptor* sesuai dengan nominal harga yang harus dibayar. Setelah pembayaran *Coin* telah mencapai nilai nominal harga yang harus dibayarkan maka sistem akan mengirim data serial menggunakan modul USB to TTL pada PC berupa data yang akan ditampilkan pada Excel.

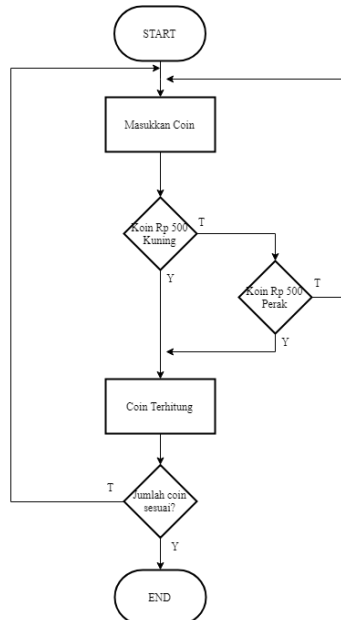
### 3.4.1 Flowchart Pembayaran

Penggunaan RTC digunakan untuk melakukan perhitungan dari tarif yang digunakan. RTC akan mulai menghitung waktu lama sewa pada saat loker digunakan. Data pada RTC akan digunakan untuk mendapatkan nilai tarif pada loker yang digunakan. Flowchart perhitungan tarif sewa menggunakan RTC adalah pada Gambar 3.7.



**Gambar 3. 7** Flowchart Perhitungan Tarif

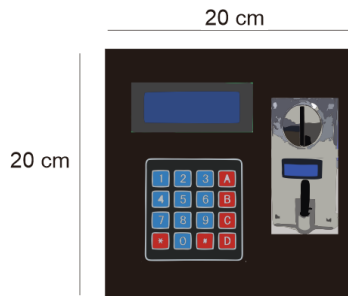
Pada alat ini dilakukan sistem pembayaran menggunakan *Coin* sebesar Rp 500 dan pada alat ini dapat membaca dua tipe *Coin* Rp 500 yaitu perak dan kuning. Flowchart pembayaran sewa loker dapat dilihat pada Gambar 3.8.



**Gambar 3. 8** Flowchart Pembayaran *Coin*

### 3.5 Desain Kubus Kontroler

Pada alat ini desain alat berupa loker berbentuk kubus dengan ukuran 20cm x 20cm x 20cm. Ketebalan dari kubus ini adalah 2mm dengan penempatan LCD 20x4 dipasang dibagian depan untuk menampilkan tampilan dari proses kerja loker, kemudian terdapat keypad sebagai *Input Password* dan *Coin Acceptor* sebagai tempat untuk pembayaran. Sedangkan pada bagian belakang terdapat lubang sebagai saluran kabel supply sebagai penyuplai daya ke rangkaian elektronika, kabel USB to TTL sebagai pengirim data serial dari Mikrokontroler ke Komputer. Untuk rancangan bentuk loker dapat dilihat pada Gambar 3.9.



**Gambar 3. 9** Desain Kubus Kontroler



**Gambar 3. 10** Kubus Kontroler Tampak Depan





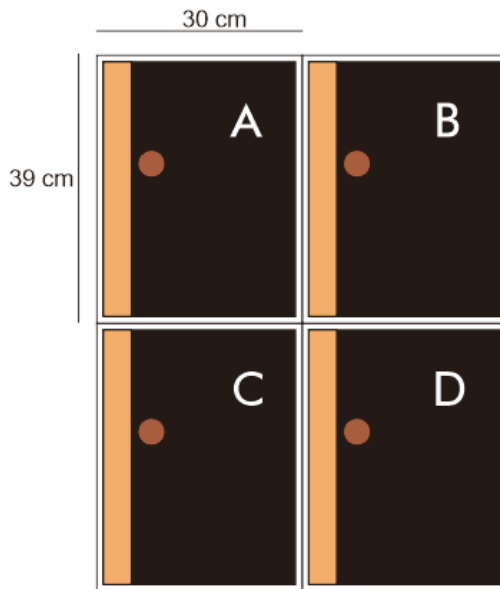
**Gambar 3. 11** Kubus Kontroler Tampak Samping



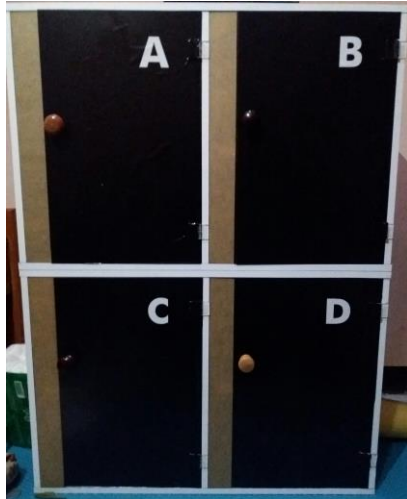
**Gambar 3. 12** Kubus Kontroler Tampak Atas

### 3.6 Desain Loker

Pada Tugas Akhir ini desain alat berupa loker sebanyak 4 buah loker berbentuk persegi yang memiliki ukuran 30cm x 39cm pada setiap lokernya. Ketebalan dari desain loker ini sebesar 1cm. Pada loker ini berfungsi sebagai penyimpanan barang yang didalamnya terdapat komponen pendukung seperti Limit Switch dan juga Solenoid Door Lock. Pintu dari loker ini terbuat dari akrilik dengan ketebalan 3mm dengan ukuran 27cm x 37cm. Loker didesain sesedemikian rupa agar dapat digunakan secara baik.



**Gambar 3. 13** Desain Loker



**Gambar 3. 14** Hasil Desain Loker Depan



**Gambar 3. 15** Hasil Desain Loker Samping



**Gambar 3. 16** Hasil Desain Loker Samping



**Gambar 3. 17** Hasil Rancangan Desain Keseluruhan

## BAB IV

### PENGUKURAN DAN PENGUJIAN

Untuk mengetahui bahwa alat telah bekerja dengan baik dan benar maka dilakukan suatu pengujian alat yang meliputi pengujian perangkat keras dan perangkat lunak. Bentuk pengujian yang dilakukan pada peralatan untuk mengetahui kesesuaian antara teori dengan hasil perancangan yaitu dengan mengetahui hasil pengukuran dari setiap peralatan yang telah dibuat.

#### 4.1 Pengukuran Rangkaian *Power Supply*

Pengukuran *power supply* dilakukan untuk mengetahui tegangan keluaran apakah tegangan yang keluar sudah sesuai. Tujuan dari pengukuran tersebut untuk mengetahui hasil tegangan yang keluar. Secara lengkap dapat dilihat pada Tabel 4.1 dibawah ini.

**Tabel 4. 1** Tabel pengukuran tegangan *Power Supply*

No	<i>Power Supply</i> 12V DC (Tanpa Beban)	Modul Regu- lator 5V DC (Tanpa Beban)	<i>Power Sup- ply</i> 12V DC	Modul Rangkaia n 5V DC
1	12.05	5.35	12.04	5.35
2	12.05	5.36	12.04	5.36
3	12.06	5.35	12.04	5.36
4	12.05	5.35	12.05	5.35
5	12.05	5.35	12.04	5.35

Dari Tabel diatas dapat dilihat bahwa tegangan *Output DC Power Supply* dapat dilihat bahwa tegangan *Output* pada *Power Supply* 12V pada saat dengan dan tanpa beban meimiiliki tegangan sebesar 11.90 – 11.92 V, sehingga tegangan *Output* dari *Power Supply* 12V dapat digunakan sebagai sumber tegangan oleh *Coin Acceptor* yang memerlukan tegangan sebesar 12V dan *Solenoid Doorlock* yang membutuhkan sumber tegangan sebesar 9-12 V. Sedangkan tegangan *Output* pada Modul Regulator 5V oada saat dengan dan tanpa beban adalah sebesar 4.97 – 5.22 sehingga

tegangan *Output* dari Modul Regulator 5V dapat digunakan sebagai sumber tegangan Mikrokontroler Atmega 16 yang tegangan kerjanya sebesar 1.8 – 5.5 V dan *Driver Relay* yang memerlukan tegangan sebesar 5 V. Selanjutnya, untuk mengetahui apakah *power supply* dapat digunakan atau tidak, dengan cara menghitung presentase *error* dengan rumus pada Persamaan 4.1 dan menghasilkan data pada Tabel 4.1 :

$$\% Error = \left| \frac{(\text{NilaiAcuan} - \text{NilaiPengukuran})}{\text{NilaiAcuan}} \right| \times 100\% \dots\dots\dots(4.1)$$

**Tabel 4. 2 Data Pengujian Power Supply**

<i>Power Supply</i>	<i>Output (V)</i>	<i>Error(%)</i>
+12 V	12,05	0,41

Pengujian dilakukan dengan mengukur keluaran dengan volt-meter, kemudian dihitung presentasi *error*.%

$$\% Error = \left| \frac{(12 - 12,05)}{12} \right| \times 100\% = 0,41 \%$$

#### **4.2 Pengujian LCD (*Liquid Crystal Display*) 20x4**

Pengujian Pengujian LCD ini dilakukan untuk mengetahui tampilan dari sistem pada alat ini bekerja sesuai sistem. Pengujian dilakukan dengan memprogram mikrokontroler ATmega 16 untuk menampilkan LCD dari sistem yang digunakan. Gambar 4.1 berikut adalah gambar tampilan dari LCD yang telah diprogram:



**Gambar 4. 1** Pengujian tampilan dari LCD

#### 4.3 Pengujian RTC DS1307

Pengujian RTC (*Real Time Clock*) DS1307 ini untuk menyesuaikan data waktu modul RTC (*Real Time Clock*) DS1307 dengan waktu sebenarnya. Pengujian dilakukan dengan cara menampilkan data waktu modul RTC (*Real Time Clock*) pada LCD dengan waktu pada komputer dengan OS Windows 10. Hasil dari pengujian dapat dilihat pada Tabel 4.3.

**Tabel 4. 3** Hasil Modul RTC (*Real Time Clock*)

Tampilan Pada LCD	Waktu Pada Laptop dengan OS Windows 10	Selisih Waktu
18:03:24	18:03:25	1 detik
18:05:39	18:05:40	1 detik
18:15:46	18:15:47	1 detik
18:23:02	18:23:03	1 detik
18:29:23	18:29:24	1 detik
18:34:12	18:34:13	1 detik
18:37:38	18:37:39	1 detik
18:39:45	18:39:46	1 detik
18:42:12	18:42:13	1 detik
18:49:15	18:49:16	1 detik

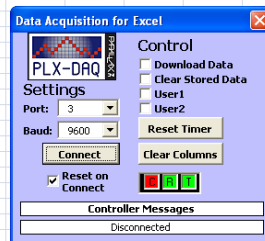
Dari data hasil pengujian pada Tabel diatas dapat dilihat bahwa waktu yang ada pada modul RTC (*Real Time Clock*)

berselisih sebesar 1 detik, dimana selisih tersebut tetap sama dari waktu ke waktu. Oleh karena itu modul RTC (*Real Time Clock*) yang digunakan sudah memiliki waktu yang sesuai dengan waktu yang sebenarnya.

#### 4.4 Pengujian Komunikasi USB to TTL Serial

Pengujian komunikasi serial USB menggunakan modul USB to TTL ini dilakukan dengan tujuan untuk menguji apakah komunikasi antara mikrokontroler ATmega 16 dengan Komputer dapat berjalan dengan baik atau tidak, hal ini sangat penting karena akan sangat berpengaruh saat mikrokontroler ATmega 16 digunakan untuk berkomunikasi dengan komputer menggunakan komunikasi dengan modul USB to TTL. Pengujian ini dilakukan menggunakan program PLX DAQ yang dapat diperoleh secara gratis dan sudah terhubung dengan Microsoft Excel. Mikrokontroler ATmega16 akan diprogram untuk berkomunikasi dengan *Personal Computer* (PC) melalui komunikasi USB to TTL dan hasil komunikasi berupa pengiriman karakter dari ATmega 16 akan dikirim ke Microsoft Excel melalui PLX DAQ dapat dilihat pada Gambar 4.2.

	A	B	C	D	E	F	G	H	I
	Connect using "PLX-DAQ Simple Test"								
1									
2	18:41:53	123456.00	DATA RITERIMA						
3	18:41:53	123456.00	DATA RITERIMA						
4	18:41:53	123456.00	DATA RITERIMA						
5	18:41:53	123456.00	DATA RITERIMA						
6	18:41:53	123456.00	DATA RITERIMA						
7	18:41:53	123456.00	DATA RITERIMA						
8	18:41:53	123456.00	DATA RITERIMA						
9	18:41:53	123456.00	DATA RITERIMA						
10	18:41:54	123456.00	DATA RITERIMA						
11	18:41:54	123456.00	DATA RITERIMA						
12	18:41:54	123456.00	DATA RITERIMA						
13	18:41:54	123456.00	DATA RITERIMA						
14	18:41:54	123456.00	DATA RITERIMA						
15	18:41:54	123456.00	DATA RITERIMA						
16	18:41:54	123456.00	DATA RITERIMA						
17	18:41:54	123456.00	DATA RITERIMA						
18	18:41:55	123456.00	DATA RITERIMA						
19	18:41:55	123456.00	DATA RITERIMA						
20	18:41:55	123456.00	DATA RITERIMA						
21	18:41:55	123456.00	DATA RITERIMA						
22	18:41:55	123456.00	DATA RITERIMA						
23	18:41:55	123456.00	DATA RITERIMA						
24	18:41:55	123456.00	DATA RITERIMA						
25	18:41:55	123456.00	DATA RITERIMA						
26	18:41:56	123456.00	DATA RITERIMA						



**Gambar 4. 2** Komunikasi Serial Menggunakan USB to TTL pada Excel



Gambar diatas dilakukan pengujian dengan mengirimkan data serial berupa tulisan 123456 DATA DITERIMA dari Mikrokontroler ke Excel dan pengiriman data dengan komunikasi serial antara mikrokontroler ATmega16 dengan PLX DAQ. Baudrate yang digunakan pada pengujian ini menggunakan 9600.

#### 4.5 Pengujian Sensor *Coin Acceptor* CH-923

Pengujian sensor *Coin Acceptor* CH-923 dilakukan untuk mengerahu bahwa sensor ini dapat mendeteksi Coin yang digunakan sesuai dengan yang teah diatur atau tidak, dimana pada alat ini yang digunakan adalah uang logam Rp 500 perak dan Rp 500 kuning. Oleh karena itu dilakukan pengujian terhadap jenis – jenis uang logam yang ada. Pengujian dengan cara memasukkan *Coin* yang berupa uang logam ke *Coin Acceptor* CH-923. Apabila jenis *Coin* yang dimasukkan sesuai, maka kebutuhan *Coin* akan ditampilkan pada *Display* yang terletak pada sisi kiri *Coin Acceptor*. Hasil pengujian terhadap *Coin* dapat dilihat pada Tabel 4.4.

**Tabel 4. 4** Hasil Pengujian *Coin Acceptor* Dengan *Coin* Rp 500 kuning

Percobaan	Tampilan pada <i>Coin Acceptor</i>	Keterangan
1	1	<i>Coin</i> Terbaca
2	1	<i>Coin</i> Terbaca
3	1	<i>Coin</i> Terbaca
4	1	<i>Coin</i> Terbaca
5	1	<i>Coin</i> Terbaca
6	1	<i>Coin</i> Terbaca
7	1	<i>Coin</i> Terbaca
8	1	<i>Coin</i> Terbaca
9	1	<i>Coin</i> Terbaca
10	1	<i>Coin</i> Terbaca

**Tabel 4. 5** Hasil Pengujian *Coin Acceptor* Dengan *Coin* Rp 500 perak

Percobaan	Tampilan pada <i>Coin Acceptor</i>	Keterangan
1	2	<i>Coin</i> Terbaca
2	2	<i>Coin</i> Terbaca
3	2	<i>Coin</i> Terbaca
4	2	<i>Coin</i> Terbaca
5	2	<i>Coin</i> Terbaca
6	2	<i>Coin</i> Terbaca
7	2	<i>Coin</i> Terbaca
8	2	<i>Coin</i> Terbaca
9	2	<i>Coin</i> Terbaca
10	2	<i>Coin</i> Terbaca

**Tabel 4. 6** Hasil Pengujian *Coin Acceptor* Dengan *Coin* Rp 200 perak

Percobaan	Tampilan pada <i>Coin Acceptor</i>	Keterangan
1	-	<i>Coin</i> Tidak Terbaca
2	-	<i>Coin</i> Tidak Terbaca
3	-	<i>Coin</i> Tidak Terbaca
4	-	<i>Coin</i> Tidak Terbaca
5	-	<i>Coin</i> Tidak Terbaca
6	-	<i>Coin</i> Tidak Terbaca
7	-	<i>Coin</i> Tidak Terbaca
8	-	<i>Coin</i> Tidak Terbaca
9	-	<i>Coin</i> Tidak Terbaca
10	-	<i>Coin</i> Tidak Terbaca

**Tabel 4. 7** Hasil Pengujian *Coin Acceptor* Dengan *Coin* Rp 1000 perak

Percobaan	Tampilan pada <i>Coin Acceptor</i>	Keterangan
1	-	<i>Coin</i> Tidak Terbaca
2	-	<i>Coin</i> Tidak Terbaca
3	-	<i>Coin</i> Tidak Terbaca
4	-	<i>Coin</i> Tidak Terbaca

Percobaan	Tampilan pada <i>Coin Acceptor</i>	Keterangan
5	-	<i>Coin</i> Tidak Terbaca
6	-	<i>Coin</i> Tidak Terbaca
7	-	<i>Coin</i> Tidak Terbaca
8	-	<i>Coin</i> Tidak Terbaca
9	-	<i>Coin</i> Tidak Terbaca
10	-	<i>Coin</i> Tidak Terbaca

Dari hasil pengujian terhadap dua jenis *Coin* berupa uang logam Rp 500 kuning dan perak dapat dilihat pada tabel – tabel diatas bahwa uang logam Rp 500 kuning menghasilkan tampilan 1 pada yang menandakan mengirim pulsa sebesar 1 gelombang dan uang logam Rp 500 perak menghasilkan tampilan dapat diterima oleh *Coin Acceptor* hal tersebut dikarenakan hanya uang logam Rp 500 kuning dan perak yang sudah dilakukan proses setting dan sampling pada *Coin Acceptor* CH-923 yang digunakan pada alat ini.

#### 4.6 Petunjuk Penggunaan Alat

Petunjuk penggunaan ditunjukkan untuk pengguna agar dapat mudah memahami cara penggunaan alat pada Tugas Akhir ini. Berikut adalah langkah langkah dalam penggunaan alat pada Tugas Akhir ini :

a) Penggunaan pada awal penggunaan loket

- Baca Petunjuk Penggunaan



**Gambar 4. 3** Petunjuk Penggunaan

- Baca kondisi loker pada LCD



**Gambar 4. 4** Kondisi Loker

- Buka pintu loker



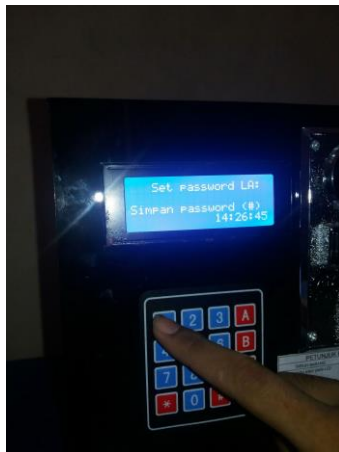
**Gambar 4. 5** Buka Loker

- Taruh barang pada loker



**Gambar 4. 6** Taruh Barang

- Set password dengan menggunakan keypad



**Gambar 4. 7** Set Password

- Password yang dimasukkan maks 4 digit  
Password terdiri dari kombinasi angka



**Gambar 4. 8** Masukkan *Password*

- Tekan (\*) untuk hapus angka



**Gambar 4. 9** Hapus *Password*

- Tekan (#) untuk simpan password



**Gambar 4. 10** Simpan *Password*

- b) Penggunaan pada saat mengambil barang

- Baca Petunjuk Penggunaan



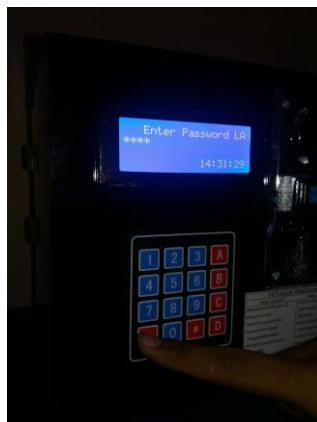
**Gambar 4. 11** Petunjuk Penggunaan

- Tekan tombol A/B/C/D pada pada *keypad* sesuai loker yang digunakan



**Gambar 4. 12** Pilih Loker

- Enter password yang telah di *set* sebelumnya



**Gambar 4. 13** *Enter Password*



- Tekan (\*) untuk hapus angka



**Gambar 4. 14** Hapus *Password*

- Tekan (#) untuk OK



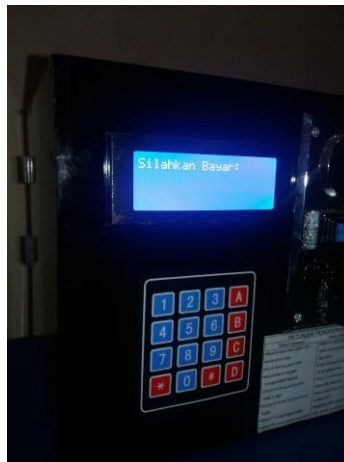
**Gambar 4. 15** Tekan #

- Akan muncul notif tarif penggunaan



**Gambar 4. 16** Notif Tarif Pembayaran

- Masukkan Coin sesuai nominal



**Gambar 4. 17** Masukkan *Coin*

- Coin senilai Rp 500 perak/kuning



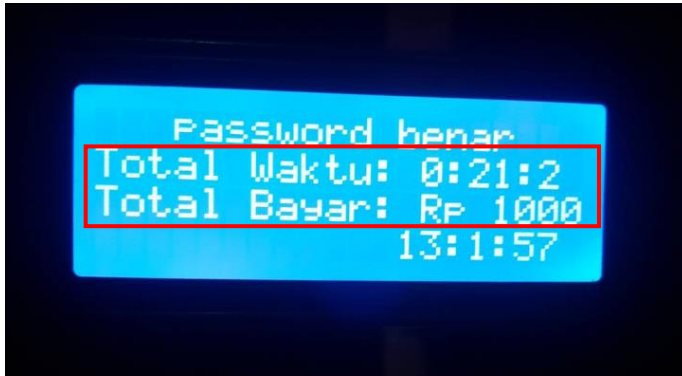
**Gambar 4. 18** Jenis *Coin*

#### **4.7 Pengujian Keseluruhan Sistem Alat**

Pada pengujian kinerja alat keseluruhan dilakukan untuk menguji tingkat keberhasilan alat ketika dioperasikan secara berulang – ulang, dimana pengujian dilakukan terhadap beberapa fitur – fitur yang ada. Bagian fitur yang diuji antara lain :

##### **4.7.1 Fitur *Timer* dan Tarif Penggunaan**

Pada pengujian terhadap fitur *Timer* dan Tarif Penggunaan dapat dikatakan berhasil ketika kebutuhan tarif harga *Coin* sesuai dengan lama penggunaan dari penyewaan loker, dimana tarif sewa loker terhitung dimulai ketika loker digunakan sehingga tarif sudah terhitung sebesar Rp 500 hingga setiap kelipatan 15 menit selanjutnya akan bertambah Rp 500. Salah satu contoh pengujian terhadap *Timer* dan Tarif Penggunaan dapat dilihat pada Gambar 4.19.

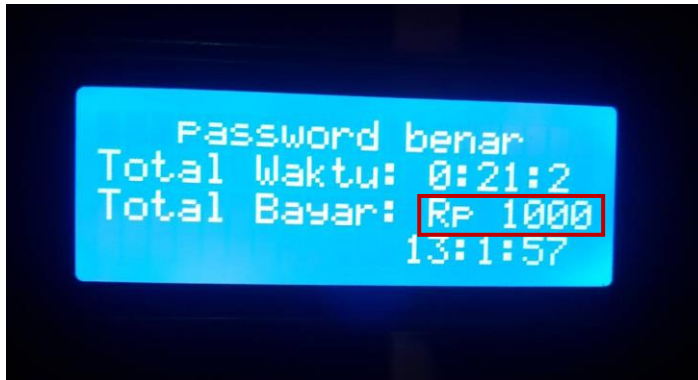


**Gambar 4. 19** Total Waktu dan Tarif Penggunaan

Pada Gambar tersebut terlihat lama penggunaan dan sesuai dari dari kebutuhan tarif *Coin*. Sehingga pada pengujian tersebut jumlah kebutuhan tarif *Coin* sesuai dengan lama penggunaan.

#### **4.7.2 Fitur Insert Coin**

Pada pengujian terhadap fitur *Insert Coin* dapat dikatakan berhasil apabila *Coin Acceptor* dapat menjalankan program yang ada apabila ditambahkan *Coin* yang sesuai yaitu *Coin* Rp 500 kuning dan *Coin* Rp 500 perak. Ketika *Coin* yang dimasukkan tidak sesuai kriteria maka tidak akan terbaca oleh sistem. Sebagai contoh proses pengujian terhadap fitur *Insert Coin*, dimana pada awalnya layar LCD (*Liquid Crystal Display*) menampilkan jumlah biaya tarif yang harus dibayarkan pengguna seperti Gambar 4.20.



**Gambar 4. 20** Jumlah Pembayaran *Coin*

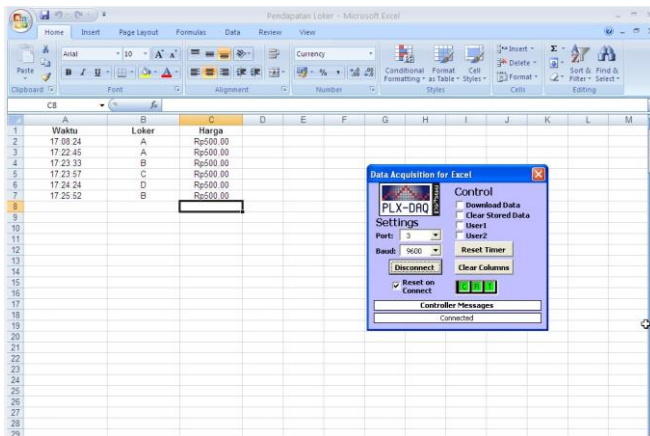
Pada saat pengguna menambahkan jenis *Coin* yang sesuai yaitu *Coin* Rp 500 kuning dan Rp 500 perak maka *Coin* pada saat membayar akan bertambah hingga mencapai biaya tarif harga yang harus dibayarkan seperti jumlah tampilan awal. Sedangkan apabila *Coin* yang ditambahkan tidak sesuai maka jumlah *Coin* tidak akan bertambah serta *Coin* yang dirambahkan akan keluar kembali dari *Coin Acceptor* seperti pada Gambar 4.21.



**Gambar 4. 21** Pembayaran *Coin*

### 4.7.3 Fitur *Serial Data*

Pada pengujian terhadap fitur *Serial Data* dapat dikatakan berhasil apabila data yang dikirim sesuai dari data penggunaan dari setiap loker. Data yang dikirim berupa waktu, loker yang digunakan, dan harga tarif loker yang telah dibayarkan. Gambar 4.22 ini adalah dari tarif loker yang telah dikirim ke Excel berupa data waktu, loker yang digunakan, dan tarif loker yang telah dibayarkan.



**Gambar 4. 22** Data Terkirim Pada Excel

Dari Gambar tersebut data yang terkirim adalah data dari waktu, loker yang digunakan, dan harga tarif loker. Data tersebut digunakan untuk mendata tarif yang telah terbayarkan dari loker yang digunakan dan mengetahui dari loker mana saja yang sering digunakan.

Untuk menentukan presentase keberhasilan terhadap kinerja alat maka dilakukan pengujian terhadap ketiga fitur secara berulang – ulang dalam pengambilan data. Ketika fitur terjadi kesalahan maka akan dianggap sebagai nilai *error* dan ketika fitur tidak terjadi kesalahan maka dianggap berhasil. Hasil pengujian dapat dilihat pada Tabel 4.8.

**Tabel 4. 8** Tabel Pengujian Kinerja Alat Keseluruhan pada Loker A

No	Pengujian Fitur			Keterangan
	<i>Timer</i> penggunaan	<i>Insert</i> <i>Coin</i>	<i>Serial Data</i>	
1	Berhasil	Berhasil	Berhasil	-
2	Berhasil	Berhasil	Berhasil	-
3	Berhasil	Berhasil	Berhasil	-
4	Berhasil	Berhasil	Berhasil	-
5	Berhasil	Berhasil	Berhasil	-
6	Berhasil	<i>Error</i>	<i>Error</i>	<i>Insert Coin</i> <i>Error</i>
7	Berhasil	Berhasil	Berhasil	-
8	Berhasil	Berhasil	Berhasil	-
9	Berhasil	Berhasil	Berhasil	-
10	Berhasil	Berhasil	Berhasil	-
11	Berhasil	Berhasil	Berhasil	-
12	Berhasil	Berhasil	Berhasil	-
13	Berhasil	Berhasil	Berhasil	-
14	Berhasil	Berhasil	Berhasil	-
15	<i>Error</i>	<i>Error</i>	<i>Error</i>	<i>Limit Switch</i> <i>Error</i>
16	Berhasil	Berhasil	Berhasil	-
17	Berhasil	Berhasil	Berhasil	-
18	Berhasil	Berhasil	Berhasil	-
19	Berhasil	Berhasil	Berhasil	-
20	Berhasil	Berhasil	Berhasil	-

**Tabel 4. 9** Tabel Pengujian Kinerja Alat Keseluruhan pada Loker B

No	Pengujian Fitur			Keterangan
	<i>Timer</i> penggunaan	<i>Insert</i> <i>Coin</i>	<i>Serial Data</i>	
1	Berhasil	Berhasil	Berhasil	-
2	Berhasil	Berhasil	Berhasil	-
3	Berhasil	Berhasil	Berhasil	-
4	Berhasil	Berhasil	Berhasil	-
5	Berhasil	Berhasil	Berhasil	-

No	<i>Timer penggunaan</i>	<i>Insert Coin</i>	<i>Serial Data</i>	Keterangan
6	Berhasil	Berhasil	Berhasil	-
7	Berhasil	Berhasil	Berhasil	-
8	<i>Error</i>	<i>Error</i>	<i>Error</i>	<i>Limit Switch Error</i>
9	Berhasil	Berhasil	Berhasil	-
10	Berhasil	Berhasil	Berhasil	-
11	Berhasil	Berhasil	Berhasil	-
12	Berhasil	Berhasil	Berhasil	-
13	Berhasil	Berhasil	Berhasil	-
14	Berhasil	Berhasil	Berhasil	-
15	Berhasil	Berhasil	Berhasil	-
16	Berhasil	Berhasil	Berhasil	-
17	Berhasil	Berhasil	Berhasil	-
18	Berhasil	Berhasil	Berhasil	-
19	Berhasil	Berhasil	Berhasil	-
20	Berhasil	Berhasil	Berhasil	-

**Tabel 4. 10** Tabel Pengujian Kinerja Alat Keseluruhan pada Loker C

No	Pengujian Fitur			Keterangan
	<i>Timer penggunaan</i>	<i>Insert Coin</i>	<i>Serial Data</i>	
1	Berhasil	Berhasil	Berhasil	-
2	Berhasil	Berhasil	Berhasil	-
3	Berhasil	Berhasil	Berhasil	-
4	<i>Error</i>	<i>Error</i>	<i>Error</i>	<i>Limit Switch Error</i>
5	Berhasil	Berhasil	Berhasil	-
6	Berhasil	Berhasil	Berhasil	-
7	Berhasil	Berhasil	Berhasil	-
8	Berhasil	Berhasil	Berhasil	-
9	Berhasil	Berhasil	Berhasil	-
10	Berhasil	Berhasil	Berhasil	-
11	Berhasil	<i>Error</i>	<i>Error</i>	<i>Insert Coin Error</i>
12	Berhasil	Berhasil	Berhasil	-



No	<i>Timer</i> penggunaan	<i>Insert</i> <i>Coin</i>	<i>Serial Data</i>	Keterangan
13	Berhasil	Berhasil	Berhasil	-
14	Berhasil	Berhasil	Berhasil	-
15	Berhasil	Berhasil	Berhasil	-
16	Berhasil	Berhasil	Berhasil	-
17	Berhasil	Berhasil	Berhasil	-
18	Berhasil	Berhasil	Berhasil	-
19	Berhasil	Berhasil	Berhasil	-
20	Berhasil	Berhasil	Berhasil	-

**Tabel 4. 11** Tabel Pengujian Kinerja Alat Keseluruhan pada Loker D

No	Pengujian Fitur			Keterangan
	<i>Timer</i> penggunaan	<i>Insert</i> <i>Coin</i>	<i>Serial Data</i>	
1	Berhasil	Berhasil	Berhasil	-
2	Berhasil	Berhasil	Berhasil	-
3	Berhasil	Berhasil	Berhasil	-
4	Berhasil	Berhasil	Berhasil	-
5	Berhasil	Berhasil	Berhasil	-
6	Berhasil	Berhasil	Berhasil	-
7	Berhasil	Berhasil	Berhasil	-
8	Berhasil	Berhasil	Berhasil	-
9	Berhasil	Berhasil	Berhasil	-
10	Berhasil	Berhasil	Berhasil	-
11	Berhasil	Berhasil	Berhasil	-
12	Berhasil	<i>Error</i>	<i>Error</i>	<i>Insert Coin</i> <i>Error</i>
13	Berhasil	Berhasil	Berhasil	-
14	Berhasil	Berhasil	Berhasil	-
15	Berhasil	Berhasil	Berhasil	-
16	Berhasil	Berhasil	Berhasil	-
17	<i>Error</i>	<i>Error</i>	<i>Error</i>	<i>Limit Switch</i> <i>Error</i>
18	Berhasil	Berhasil	Berhasil	-
19	Berhasil	Berhasil	Berhasil	-
20	Berhasil	Berhasil	Berhasil	-

Dari hasil pengujian kinerja alat secara keseluruhan yang dilakukan sebanyak 20 kali dari setiap loker A, B, C, dan D bahwa terjadi tujuh kali *error*. *Error* yang terjadi disebabkan dari *Limit Switch* dan *Insert Coin*. Dari hasil pengujian tersebut presentase keberhasilan  $(73/80) \times 100\% = 91,25\%$  dengan presentase *error* sebesar  $100\% - 91,25\% = 8,75\%$

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Pada penelitian kali ini, dapat diambil kesimpulan antara lain sebagai berikut:

1. Tegangan *Output* yang dihasilkan oleh *Power Supply* pada saat terpasang beban lebih kecil 0,01V dari pada Tegangan *Output* pada saat tidak terpasang.
2. Output tegangan dari *Power Supply* 12V tanpa menggunakan beban sebesar 12.05 V – 12.06 V
3. Pada pengujian terhadap jenis – jenis uang logam yang ada terhadap sensor *Coin Acceptor* CH-923. Uang logam yang dapat digunakan dan terdeteksi oleh sensor tersebut adalah uang logam Rp 500 kuning dan Rp 500 perak. Hal tersebut dikarenakan kedua jenis uang logam tersebut sudah dilakukan proses *setting* dan *sampling* pada sensor *Coin Acceptor* CH-923.
4. Pada pengujian tampilan serial monitor dapat terkirim dari komunikasi Mikrokontroler dengan Komputer berupa data pada Excel.
5. Pada pengujian RTC DS1307 memiliki selisih waktu sebesar 1 detik dengan waktu pada Komputer, dimana selisih tersebut tetap sama dari waktu ke waktu.
6. *Input Coin Acceptor* dan tampilan waktu RTC DS1307 dapat ditampilkan pada LCD.
7. Pada pengujian secara keseluruhan dengan mencoba seluruh fitur yang tersedia didapatkan nilai presentase *error* sebesar 8,75% sedangkan untuk presentase nilai keberhasilan 91,25%.

#### **5.2 Saran**

Saran untuk pengembangan alat sejenis selanjutnya yaitu dapat dilakukan dengan cara menambahkan beberapa jenis uang logam yang bisa digunakan dan juga dapat menambahkan fitur monitoring online sehingga dapat mengetahui lama penggunaan loker yang dapat diakses secara jarak jauh sehingga tidak akan terjadi kelebihan lama sewa waktu yang berpengaruh pada biaya tarif penggunaan dan lebih terjaga tingkat keamanannya.

-----Halaman ini sengaja dikosongkan-----

## DAFTAR PUSTAKA

Andrianto, Heri. 2015. "Pemrograman Mikrokontroler AVR AT-Mega16 Menggunakan Bahasa C (CodeVision AVR)", Bandung: Informatika Bandung.

Budiharto, Widodo. 2008. "Panduan Praktikum Mikrokontroler AVR ATmega16". Jakarta: Elex Media Komputindo.

Dhananjay, Gadre. 2001. "*Programing and Customizing The AVR Microcontroller*". New York: McGraw-Hill.

Eko putra, Agfianto. 2002. "Belajar Mikrokontroler ATmega16". Yogyakarta: Gaya Media.

Nugraha Gerri, Irman. 2013. "Perancangan dan Implementasi Alat Parkir Sepeda Dengan Sistem Pembayaran Otomatis Menggunakan *Coin Acceptor*". UNIKOM Bandung: Tidak diterbitkan.

Morton, John. 2007. "*AVR an Introductory Course*". England: Newnes.

Perdana, Latif. 2012. "Modul AVR *Trainer*". Yogyakarta: Mercubuana,.

Setiawan, Iwan. 2006. "*Tutorial Microcontroller AVR Part I*". Semarang: UNDIP.

Suroto. 2000. "Strategi pembangunan dan Perencanaan Perencanaan Kesempatan Kerja". Yogyakarta: Universitas Gajah Mada.

Team IE. 2006. "Panduan Praktis Mikrokontroler Keluarga AVR". Surabaya: Innovative Electronic.

Wardana, Lingga. 2006. "Belajar Sendiri Mikrokontroler AVR Seri ATmega16, Simulasi *Hardware* dan Aplikasi". Yogyakarta: Andi.

-----Halaman ini sengaja dikosongkan-----

## LAMPIRAN

```
#include <mega16.h>
#include <delay.h>
#include <stdlib.h>

#define ls1 PIND.6
#define ls2 PIND.7
#define ls3 PINC.5
#define ls4 PINC.4
#define door_lock1 PORTC.3
#define door_lock2 PORTC.2
#define door_lock3 PORTC.1
#define door_lock4 PORTC.0

// I2C Bus functions

// I2C Bus functions
#include <i2c.h>

// DS1307 Real Time Clock functions
#include <ds1307.h>

// Alphanumeric LCD functions
#include <alcd.h>

// Declare your global variables here

#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];
```

```

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index=0,rx_rd_index=0;
#else
unsigned int rx_wr_index=0,rx_rd_index=0;
#endif

#if RX_BUFFER_SIZE < 256
unsigned char rx_counter=0;
#else
unsigned int rx_counter=0;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
    char status,data;
    status=UCSRA;
    data=UDR;
    if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
    {
        rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
        // special case for receiver buffer size=256
        if (++rx_counter == 0) rx_buffer_overflow=1;
#else
        if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
        if (++rx_counter == RX_BUFFER_SIZE)
        {
            rx_counter=0;
            rx_buffer_overflow=1;
        }
#endif
    }
}

```



```

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index++];
#if RX_BUFFER_SIZE != 256
if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#endif
asm("cli")
--rx_counter;
asm("sei")
return data;
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

long int nilai=0,nilai2=0,array[10];
int i,a,flag=0,flag2=0;
char temp[32];
unsigned char jam, menit, detik, tanggal, bulan, tahun;
int l1,l2,l3,l4,d=0,e=0;
eeprom int status[5];
eeprom int password[4],correct_password[e];
int timer,data_key,bayar_detik,bayar_menit,bayar_jam;
int prev_jam[4],prev_menit[4],prev_detik[4];
int prev_jam2[4],prev_menit2[4],prev_detik2[4];
float freq; // to store value of frequency value
unsigned int u=0,dur; //i=number of overflows in one second
// dur to store the value of TCNT1 register
char buffer[8]; // to store the frequency value as a string to be
displayed on lcd
int counter_duit=0,bayar_duit=0;

```

```

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
    u++;
}

void baca_ls(){
    if(ls1==0){l1=0;}
    else{l1=1;d=1;e=1;}
    if(ls2==0){l2=0;}
    else{l2=1;d=2;e=2;}
    if(ls3==0){l3=0;}
    else{l3=1;d=3;e=3;}
    if(ls4==0){l4=0;}
    else{l4=1;d=4;e=4;}
}

void waktu(){
    //lcd_clear();
    rtc_get_time(&jam,&menit,&detik);
    lcd_gotoxy(12,3);
    sprintf(temp,"%d:%d:%d ", jam, menit, detik);
    lcd_puts(temp);
}

void lock(){
    //digitalWrite(solenoid,HIGH);
}

void unlocked(){
    //digitalWrite(solenoid,LOW);
}

void simpan_variabel(){
    if (i==1){
        nilai=array[i];
        lcd_gotoxy(i-1,1);
        sprintf(temp,"%d",array[i]);
        lcd_puts(temp);
        password[e]=nilai;
        correct_password[e]=password[e];
    }
}

```

```

        //status[d]=1;
        delay_ms(500);
        lcd_gotoxy(i-1,1);
        lcd_putsf("*");
    }
    if (i>=2 && i<=4){
        nilai=(nilai*10)+array[i];
        lcd_gotoxy(i-1,1);
        sprintf(temp,"%d",array[i]);
        lcd_puts(temp);
        password[e]=nilai;
        correct_password[e]=password[e];
        delay_ms(500);
        lcd_gotoxy(i-1,1);
        lcd_putsf("*");
    }
}

void simpan_variabel2(){
    if (i==1){
        nilai2=array[i];
        lcd_gotoxy(i-1,1);
        sprintf(temp,"%d",array[i]);
        lcd_puts(temp);
        delay_ms(500);
        lcd_gotoxy(i-1,1);
        lcd_putsf("*");
    }
    if (i>=2 && i<=4){
        nilai2=(nilai2*10)+array[i];
        lcd_gotoxy(i-1,1);
        sprintf(temp,"%d",array[i]);
        lcd_puts(temp);
        delay_ms(500);
        lcd_gotoxy(i-1,1);
        lcd_putsf("*");
    }
}

void hapus(){

```

```

        //c++;
        nilai=(nilai-array[i])/10;
        //itoa(nilai,temp);
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putsf("enter password");
        for(a=0;a<i;a++){
            lcd_gotoxy(a,1);
            lcd_putsf("*");
        }
        lcd_gotoxy(7,1);
        sprintf(temp,"%d",nilai);
        lcd_puts(temp);
    }
    void enter(){
        i=0; flag=0; d=0;
        status[e]=1;
        password[e]=nilai;
        prev_jam[e]=jam;
        prev_menit[e]=menit;
        prev_detik[e]=detik;
        delay_ms(100);
        lcd_clear();
        delay_ms(500);
        lcd_gotoxy(3,0);
        lcd_putsf("Password Saved");
        lcd_gotoxy(0,1);
        sprintf(temp,"%d",nilai);
        lcd_puts(temp);
        delay_ms(1000);
        if(status[1]==1){
            door_lock1=0;
        }
        if(status[2]==1){
            door_lock2=0;
        }
        if(status[3]==1){
            door_lock3=0;
        }
    }

```

```

        if(status[4]==1){
            door_lock4=0;
        }
    }
    void enter2(){

        prev_jam2[e]=jam;
        prev_menit2[e]=menit;
        prev_detik2[e]=detik;
        if(correct_password[e]==nilai2){
            bayar_detik=prev_detik2[e]-prev_detik[e];
            bayar_menit=prev_menit2[e]-prev_menit[e];
            bayar_jam=prev_jam2[e]-prev_jam[e];
            if(bayar_detik<0){
                bayar_detik=(prev_detik2[e]+60)-prev_detik[e];
                bayar_menit=bayar_menit-1;
            }
            if(bayar_menit<0){
                bayar_menit=(prev_menit2[e]+60)-prev_menit[e];
                bayar_jam=bayar_jam-1;
            }
            if(bayar_jam<0){
                bayar_jam=(prev_jam2[e]+60)-prev_jam[e]-1;
            }
            delay_ms(100);
            lcd_clear();
            delay_ms(250);
            lcd_gotoxy(3,0);
            lcd_putsf("password benar");
            lcd_gotoxy(0,1);
            lcd_putsf("Total Waktu: ");
            sprintf(temp,"%d:%d:%d",
            ",bayar_jam,bayar_menit,bayar_detik);
            bayar_menit=((bayar_menit/15) *500)+500;
            lcd_puts(temp);
            lcd_gotoxy(0,2);
            lcd_putsf("Total Bayar: ");
            sprintf(temp,"Rp %d ",bayar_menit);
            lcd_puts(temp);

```

```

        delay_ms(1500);
        i=0; flag2=5; d=0;
        status[e]=0;
    }
    else{
        delay_ms(100);
        lcd_clear();
        delay_ms(250);
        lcd_gotoxy(3,0);
        lcd_putsf("password salah");
        delay_ms(1500);
        flag=0;
        i=0;
        status[e]=1;
        d=0;
    }
}

void keypad(){
    //lcd_gotoxy(0,0);
    //lcd_putsf("Enter Password");

    PORTA = 0b11110111;
    delay_ms(50);
    if(PINA.7==0){data_key=44; delay_ms(50);} //D
    else if(PINA.6==0){ enter(); delay_ms(50);}
    else if(PINA.5==0){ i++; array[i]=0; sim-
pan_variabel();delay_ms(50);}
    else if(PINA.4==0){ i--; hapus(); delay_ms(50);}

    PORTA = 0b11111011;
    delay_ms(50);
    if(PINA.7==0){data_key=33; delay_ms(50);} //C
    else if(PINA.6==0){i++; array[i]=9; simpan_variabel(); de-
lay_ms(50);}
    else if(PINA.5==0){i++; array[i]=8; simpan_variabel(); de-
lay_ms(50);}
    else if(PINA.4==0){i++; array[i]=7; simpan_variabel(); de-
lay_ms(50);}

```

```

    PORTA = 0b11111101;
    delay_ms(50);
    if(PINA.7==0){data_key=22; delay_ms(50);} //B
    else if(PINA.6==0){i++; array[i]=6; simpan_variabel(); de-
lay_ms(50);}
    else if(PINA.5==0){i++; array[i]=5; simpan_variabel(); de-
lay_ms(50);}
    else if(PINA.4==0){i++; array[i]=4; simpan_variabel(); de-
lay_ms(50);}

```

```

    PORTA = 0b11111110;
    delay_ms(50);
    if(PINA.7==0){data_key=11; delay_ms(50);} //A
    else if(PINA.6==0){i++; array[i]=3; simpan_variabel(); de-
lay_ms(50);}
    else if(PINA.5==0){i++; array[i]=2; simpan_variabel(); de-
lay_ms(50);}
    else if(PINA.4==0){i++; array[i]=1; simpan_variabel(); de-
lay_ms(50);}

```

```

    if(i>=4){i=4;}
}
void keypad2(){
    //lcd_gotoxy(0,0);
    //lcd_putsf("Enter Password");
}

```

```

    PORTA = 0b11110111;
    delay_ms(50);
    if(PINA.7==0){data_key=44; delay_ms(50);} //D
    else if(PINA.6==0){ enter2(); delay_ms(50);}
    else if(PINA.5==0){ i++; array[i]=0; sim-
pan_variabel2();delay_ms(50);}
    else if(PINA.4==0){ i--; hapus(); delay_ms(50);}

```

```

    PORTA = 0b11111011;
    delay_ms(50);
    if(PINA.7==0){data_key=33; delay_ms(50);} //C
    else if(PINA.6==0){i++; array[i]=9; simpan_variabel2(); de-
lay_ms(50);}

```

```

        else if(PINA.5==0){i++; array[i]=8; simpan_variabel2(); de-
lay_ms(50);}
        else if(PINA.4==0){i++; array[i]=7; simpan_variabel2(); de-
lay_ms(50);}

        PORTA = 0b11111101;
        delay_ms(50);
        if(PINA.7==0){data_key=22; delay_ms(50);} //B
        else if(PINA.6==0){i++; array[i]=6; simpan_variabel2(); de-
lay_ms(50);}
        else if(PINA.5==0){i++; array[i]=5; simpan_variabel2(); de-
lay_ms(50);}
        else if(PINA.4==0){i++; array[i]=4; simpan_variabel2(); de-
lay_ms(50);}

        PORTA = 0b11111110;
        delay_ms(50);
        if(PINA.7==0){data_key=11; delay_ms(50);} //A
        else if(PINA.6==0){i++; array[i]=3; simpan_variabel2(); de-
lay_ms(50);}
        else if(PINA.5==0){i++; array[i]=2; simpan_variabel2(); de-
lay_ms(50);}
        else if(PINA.4==0){i++; array[i]=1; simpan_variabel2(); de-
lay_ms(50);}
    }
    void baca_coin(){
        TIMSK=0x04;
        TCCR1B=0x07;
        delay_ms(1000);
        TCCR1B=0x00;
        TIMSK=0x00;
        dur=TCNT1;
        //freq = (((dur + i*65536)*60)/4.8)*0.0166;
        TCNT1=0x0000;
        i=0;
        //lcd_gotoxy(0,0);
        ///lcd_putsf("Coin=");
        //lcd_gotoxy(0,1);
        //ftoa(dur,2,buffer);

```



```

        //lcd_puts(buffer);
        //lcd_putsf("M");
    }

    void main(void)
    {
        // Declare your local variables here

        // Input/Output Ports initialization
        // Port A initialization
        // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
        Bit1=In Bit0=In
        DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4)
        | (1<<DDA3) | (1<<DDA2) | (1<<DDA1) | (1<<DDA0);
        // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T
        Bit0=T
        PORTA=(1<<PORTA7) | (1<<PORTA6) | (1<<PORTA5) |
        (1<<PORTA4) | (1<<PORTA3) | (1<<PORTA2) | (1<<PORTA1) |
        (1<<PORTA0);

        // Port B initialization
        // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
        Bit1=In Bit0=In
        DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) |
        (0<<DDB3) | (0<<DDB2) | (0<<DDB1) | (0<<DDB0);
        // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T
        Bit0=T
        PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) |
        (0<<PORTB4) | (0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) |
        (0<<PORTB0);
        //PORTB=0xFF;
        //DDRB=0x0F;
        // Port C initialization
        // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
        Bit1=In Bit0=In
        DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) |
        (1<<DDC3) | (1<<DDC2) | (1<<DDC1) | (1<<DDC0);
        // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T
        Bit0=T

```

```

PORTC=(1<<PORTC7) | (1<<PORTC6) | (1<<PORTC5) |
(1<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) |
(0<<PORTC0);

```

```

//PORTC=0xF0;

```

```

//DDRC=0x00;

```

```

// Port D initialization

```

```

// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In
Bit1=In Bit0=In

```

```

DDRD=(0<<DDD7) | (0<<DDD6) | (0<<DDD5) | (0<<DDD4)
| (0<<DDD3) | (0<<DDD2) | (0<<DDD1) | (0<<DDD0);

```

```

// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T
Bit0=T

```

```

PORTD=(1<<PORTD7) | (1<<PORTD6) | (0<<PORTD5) |
(0<<PORTD4) | (0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) |
(0<<PORTD0);

```

```

// Timer/Counter 0 initialization

```

```

// Clock source: System Clock

```

```

// Clock value: Timer 0 Stopped

```

```

// Mode: Normal top=0xFF

```

```

// OC0 output: Disconnected

```

```

TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) |
(0<<WGM01) | (0<<CS02) | (0<<CS01) | (0<<CS00);

```

```

TCNT0=0x00;

```

```

OCR0=0x00;

```

```

// Timer/Counter 1 initialization

```

```

// Clock source: System Clock

```

```

// Clock value: Timer1 Stopped

```

```

// Mode: Normal top=0xFFFF

```

```

// OC1A output: Disconnected

```

```

// OC1B output: Disconnected

```

```

// Noise Canceler: Off

```

```

// Input Capture on Falling Edge

```

```

// Timer1 Overflow Interrupt: On

```

```

// Input Capture Interrupt: Off

```

```

// Compare A Match Interrupt: Off

```

```

// Compare B Match Interrupt: Off
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) |
(0<<COM1B0) | (0<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) |
(0<<WGM12) | (0<<CS12) | (0<<CS11) | (0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0<<AS2;
TCCR2=(0<<PWM2) | (0<<COM21) | (0<<COM20) |
(0<<CTC2) | (0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) |
(0<<OCIE1A) | (0<<OCIE1B) | (1<<TOIE1) | (0<<OCIE0) |
(0<<TOIE0);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) |
(0<<ISC00);
MCUCSR=(0<<ISC2);

// USART initialization

```

```

// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) |
(0<<DOR) | (0<<UPE) | (0<<U2X) | (0<<MPCM);
UCSRB=(1<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) |
(1<<RXEN) | (1<<TXEN) | (0<<UCSZ2) | (0<<RXB8) |
(0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) |
(0<<UPM0) | (0<<USBS) | (1<<UCSZ1) | (1<<UCSZ0) |
(0<<UCPOL);
UBRRH=0x00;
UBRRL=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) |
(0<<ACIE) | (0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);
SFIOR=(0<<ACME);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) |
(0<<ADIF) | (0<<ADIE) | (0<<ADPS2) | (0<<ADPS1) |
(0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) |
(0<<CPOL) | (0<<CPHA) | (0<<SPR1) | (0<<SPR0);

// TWI initialization
// TWI disabled

```

```
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) |
(0<<TWEN) | (0<<TWIE);
```

```
// Bit-Banged I2C Bus initialization
// I2C Port: PORTD
// I2C SDA bit: 5
// I2C SCL bit: 4
// Bit Rate: 100 kHz
// Note: I2C settings are specified in the
// Project|Configure|C Compiler|Libraries|I2C menu.
i2c_init();
```

```
// DS1307 Real Time Clock initialization
// Square wave output on pin SQW/OUT: Off
// SQW/OUT pin state: 0
rtc_init(0,0,0);
```

```
// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD
menu:
```

```
// RS - PORTB Bit 0
// RD - PORTB Bit 2
// EN - PORTB Bit 3
// D4 - PORTB Bit 4
// D5 - PORTB Bit 5
// D6 - PORTB Bit 6
// D7 - PORTB Bit 7
// Characters/line: 16
lcd_init(20);
//rtc_set_time(14,51,30); //mengeset jam 18:18:30
printf("CLEARDATA\n"); //clears up any data left from previ-
ous projects
printf("LABEL,Time,Delay,Jam,Menit,Detik,Total,\n");
//always write LABEL, so excel knows the next things will be the
names of the columns (instead of Acolumn you could write Time for
instance)
printf("RESETTIMER\n"); //resets timer to 0
// Global enable interrupts
```

```

door_lock1=1;
door_lock2=1;
door_lock3=1;
door_lock4=1;
#asm("sei")

while (1){
    start:
    while(flag==0){
        keypad();
        waktu();
        baca_ls();
        lcd_gotoxy(0,0);
        lcd_putsf("LokA:");
        //sprintf(temp,"%d ",status[1]);
        //lcd_puts(temp);
        lcd_gotoxy(0,1);
        lcd_putsf("LokB:");
        //sprintf(temp,"%d ",status[2]);
        //lcd_puts(temp);
        lcd_gotoxy(0,2);
        lcd_putsf("LokC:");
        //sprintf(temp,"%d ",status[3]);
        //lcd_puts(temp);
        lcd_gotoxy(0,3);
        lcd_putsf("LokD:");
        //sprintf(temp,"%d ",status[4]);
        //lcd_puts(temp);
        if(status[1]==1){
            lcd_gotoxy(5,0);
            lcd_putsf("penuh ");
            door_lock1=0;
        }
        else{
            lcd_gotoxy(5,0);
            lcd_putsf("kosong");
            door_lock1=1;
        }
        if(status[2]==1){

```

```

        lcd_gotoxy(5,1);
        lcd_putsf("penuh ");
        door_lock2=0;
    }
    else{
        lcd_gotoxy(5,1);
        lcd_putsf("kosong");
        door_lock2=1;
    }
    if(status[3]==1){
        lcd_gotoxy(5,2);
        lcd_putsf("penuh ");
        door_lock3=0;
    }
    else{
        lcd_gotoxy(5,2);
        lcd_putsf("kosong");
        door_lock3=1;
    }
    if(status[4]==1){
        lcd_gotoxy(5,3);
        lcd_putsf("penuh ");
        door_lock4=0;
    }
    else{
        lcd_gotoxy(5,3);
        lcd_putsf("kosong");
        door_lock4=1;
    }
    if(d==1){
        delay_ms(50);
        flag=1;
        lcd_clear();
        delay_ms(250);
        lcd_gotoxy(3,0);
        lcd_putsf("Set password LA:");
        lcd_gotoxy (0,2);
        lcd_putsf ("Simpan password (#)");
        while(flag==1){

```

```

        for(timer=0;timer<70;timer++){
            keypad();
            waktu();
            delay_ms(20);
        }
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
}
else if(d==2){
    delay_ms(50);
    flag=1;
    lcd_clear();
    delay_ms(250);
    lcd_gotoxy(3,0);
    lcd_putsf("Set password LB:");
    lcd_gotoxy(0,2);
    lcd_putsf("Simpan password (#)");
    while(flag==1){
        for(timer=0;timer<70;timer++){
            keypad();
            waktu();
            delay_ms(20);
        }
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
}
else if(d==3){
    delay_ms(50);
    flag=1;
    lcd_clear();
    delay_ms(250);
    lcd_gotoxy(3,0);
    lcd_putsf("Set password LC:");

```



```

    lcd_gotoxy (0,2);
    lcd_putsf ("Simpan password (#)");
    while(flag==1){
        for(timer=0;timer<70;timer++){
            keypad();
            waktu();
            delay_ms(20);
        }
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
}
else if(d==4){
    delay_ms(50);
    flag=1;
    lcd_clear();
    delay_ms(250);
    lcd_gotoxy(3,0);
    lcd_putsf("Set password LD:");
    lcd_gotoxy (0,2);
    lcd_putsf ("Simpan password (#)");
    while(flag==1){
        for(timer=0;timer<70;timer++){
            keypad();
            waktu();
            delay_ms(20);
        }
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
}
}
if(data_key==11){
    flag=2;
    e=1;
    delay_ms(100);
}

```

```

lcd_clear();
lcd_gotoxy(3,0);
lcd_putsf("Enter Password LA");
while(flag==2){
    for(timer=0;timer<40;timer++){
        keypad2();
        waktu();
        delay_ms(20);
    }

    data_key=0;
    //status[e]=0;
    i=0;
    flag=0;
    d=0;
    lcd_clear();
}
while(flag2==5){
    delay_ms(50);
    lcd_gotoxy(0,0);
    lcd_putsf("Silahkan Bayar:  ");
    baca_coin();
    if(dur==2 || dur==1){
        delay_ms(50);
        lcd_clear();
        delay_ms(100);
        counter_duit++;
        bayar_duit=counter_duit*500;
        lcd_gotoxy(0,1);
        sprintf(temp,"%d ",bayar_duit);
        lcd_puts(temp);
        if(bayar_duit>=bayar_menit){
            door_lock1=1;
            delay_ms(200);
            lcd_gotoxy(0,2);
            lcd_putsf("Cukup!!!");
            printf("DATA,TIME,"); //writes the time in the

```

first column A and the time since the measurements started in column B

```

        printf("A");
        printf(",");
        printf("%d", bayar_duit);
        printf("\n");
        delay_ms(1500);
        i=0;
        lcd_clear();
        bayar_duit=0;
        counter_duit=0;
        flag2=0;
        flag=0;
        d=0;
    }
}

}
}
else if(data_key==22){
    flag=2;
    e=2;
    delay_ms(100);
    lcd_clear();
    lcd_gotoxy(3,0);
    lcd_putsf("Enter Password LB");
    while(flag==2){
        for(timer=0;timer<40;timer++){
            keypad2();
            waktu();
            delay_ms(20);
        }
        //i=0;
        //flag=0;
        //d=0;
        data_key=0;
        //status[e]=0;
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
}

```

```

    }
    while(flag2==5){
        delay_ms(50);
        lcd_gotoxy(0,0);
        lcd_putsf("Silahkan Bayar:  ");
        baca_coin();
        if(dur==2 || dur==1){
            delay_ms(50);
            lcd_clear();
            delay_ms(100);
            counter_duit++;
            bayar_duit=counter_duit*500;
            lcd_gotoxy(0,1);
            sprintf(temp,"%d ",bayar_duit);
            lcd_puts(temp);
            if(bayar_duit>=bayar_menit){
                door_lock2=1;
                delay_ms(200);
                lcd_gotoxy(0,2);
                lcd_putsf("Cukup!!!");
                printf("DATA,TIME,"); //writes the time in the
first column A and the time since the measurements started in col-
umn B

                printf("B");
                printf(",");
                printf("%d",bayar_duit);
                printf("\n");
                delay_ms(1500);
                i=0;
                lcd_clear();
                bayar_duit=0;
                counter_duit=0;
                flag2=0;
                flag=0;
                d=0;
            }
        }
    }
}

```

```

}
else if(data_key==33){
    flag=2;
    e=3;
    delay_ms(100);
    lcd_clear();
    lcd_gotoxy(3,0);
    lcd_putsf("Enter Password LC");
    while(flag==2){
        for(timer=0;timer<40;timer++){
            keypad2();
            waktu();
            delay_ms(20);
        }
        //i=0;
        //flag=0;
        //d=0;
        data_key=0;
        //status[e]=0;
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
    while(flag2==5){
        delay_ms(50);
        lcd_gotoxy(0,0);
        lcd_putsf("Silahkan Bayar:  ");
        baca_coin();
        if(dur==2 || dur==1){
            delay_ms(50);
            lcd_clear();
            delay_ms(100);
            counter_duit++;
            bayar_duit=counter_duit*500;
            lcd_gotoxy(0,1);
            sprintf(temp,"%d ",bayar_duit);
            lcd_puts(temp);
            if(bayar_duit>=bayar_menit){

```

```

        door_lock3=1;
        delay_ms(200);
        lcd_gotoxy(0,2);
        lcd_putsf("Cukup!!!");
        printf("DATA,TIME,"); //writes the time in the
first column A and the time since the measurements started in col-
umn B

```

```

        printf("C");
        printf(",");
        printf("%d",bayar_duit);
        printf("\n");
        delay_ms(1500);
        i=0;
        lcd_clear();
        bayar_duit=0;
        counter_duit=0;
        flag2=0;
        flag=0;
        d=0;
    }
}

}
else if(data_key==44){
    flag=2;
    e=4;
    delay_ms(100);
    lcd_clear();
    lcd_gotoxy(3,0);
    lcd_putsf("Enter Password LD");
    while(flag==2){
        for(timer=0;timer<40;timer++){
            keypad2();
            waktu();
            delay_ms(20);
        }
        //i=0;
        //flag=0;
    }
}

```

```

        //d=0;
        data_key=0;
        //status[e]=0;
        i=0;
        flag=0;
        d=0;
        lcd_clear();
    }
    while(flag2==5){
        delay_ms(50);
        lcd_gotoxy(0,0);
        lcd_putsf("Silahkan Bayar:  ");
        baca_coin();
        if(dur==2 || dur==1){
            delay_ms(50);
            lcd_clear();
            delay_ms(100);
            counter_duit++;
            bayar_duit=counter_duit*500;
            lcd_gotoxy(0,1);
            sprintf(temp,"%d ",bayar_duit);
            lcd_puts(temp);
            if(bayar_duit>=bayar_menit){
                door_lock4=1;
                delay_ms(200);
                lcd_gotoxy(0,2);
                lcd_putsf("Cukup!!!");
                printf("DATA,TIME,"); //writes the time in the
first column A and the time since the measurements started in col-
umn B

                printf("D");
                printf(",");
                printf("%d",bayar_duit);
                printf("\n");
                delay_ms(1500);
                i=0;
                lcd_clear();
                bayar_duit=0;
                counter_duit=0;

```

```
        flag2=0;
        flag=0;
        d=0;
    }
}
}
}
}
```



## RIWAYAT HIDUP PENULIS



Nama : Ega Hasbi Rizqullah  
TTL : Mojokerto, 3 Juli 1997  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat : Jl. Malabar 3 No 2  
Kedundung Indah Kota  
Mojokerto  
Telp/HP : 085731678089  
E-mail : egahasbi@gmail.com

### RIWAYAT PENDIDIKAN:

1. 2003-2009 : SD Negeri Wates VI Kota Mojokerto
2. 2009-2012 : SMP Negeri 2 Kota Mojokerto
3. 2012-2015 : SMA Negeri 2 Kota Mojokerto
4. 2015-2018 : Program Studi Komputer Kontrol,  
Departemen Teknik Elektro Otomasi,  
Fakultas Vokasi, Institut Teknologi  
Sepuluh Nopember, Surabaya

### PENGALAMAN KERJA

Kerja Praktek di PT Indofood Sukses Makmur Tbk. Divisi  
Bogasari Surabaya

### PENGALAMAN ORGANISASI

1. KESMA HIMAD3TEKTRO 2016/2017
2. KESMA HIMAD3TEKTRO 2017/2018

-----Halaman ini sengaja dikosongkan-----